



A nonlinear PDE model for reconstructing a regular surface from sampled data using a level set formulation on triangular meshes.

Alexandra Claisse, Pascal Frey

► To cite this version:

Alexandra Claisse, Pascal Frey. A nonlinear PDE model for reconstructing a regular surface from sampled data using a level set formulation on triangular meshes.. 2011. <hal-00559751>

HAL Id: hal-00559751

<https://hal.archives-ouvertes.fr/hal-00559751>

Submitted on 26 Jan 2011

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

A nonlinear PDE model for reconstructing a regular surface from sampled data using a level set formulation on triangular meshes.

A. Claisse^a, P. Frey^a

^a*UPMC Univ Paris 06, UMR 7598, Laboratoire J.-L. Lions, F-75005 Paris, France.*

Abstract

In this paper, we propose a nonlinear PDE model for reconstructing a regular surface from sampled data. At first, we show the existence and the uniqueness of a viscosity solution to this problem. Then we propose a numerical scheme for solving the nonlinear level set equation on unstructured triangulations adapted to the data sample. We show the consistency of this scheme. In addition, we show how to compute nodewise first and second order derivatives. Some application examples of curve or surface construction are provided to illustrate the potential and to demonstrate the accuracy of this method.

Keywords: numerical analysis, level set method, mean curvature evolution, Hamilton-Jacobi equation, nonlinear PDE problem, surface reconstruction, unstructured mesh.

1. Introduction

In this paper, we address the reconstruction of a smooth closed manifold surface, that may consists of several connected components, from a sample V of unstructured points. This dataset is typically supplied by three dimensional acquisition devices and points coordinates are possibly noisy or inaccurate. For this reason, the points are supposed to be lying on a surface and not arbitrarily distributed in \mathbb{R}^3 . Following the pioneering work of Osher and Sethian on tracking dynamic interfaces [37], this problem is formulated here such that the desired surface is the solution of an evolutionary system of nonlinear partial differential equations endowed an initial value condition (Section 3). This model is implemented using the level set curvature flow scheme that provides several key advantages and requires only a stopping criterion related to a distance check (Section 4). Like many related works (see Section 2 for a brief survey), the surface is embedded as the zero isocontour of a higher dimensional function u , and its motion can be derived by solving an equivalent equation of motion written for u . This implicit representation of the surface offers topological flexibility. Moreover, efficient algorithms are available for converting this surface into a triangulation,

*corresponding author: A. Claisse

**The first author has been supported by a research grant from the Région Ile-de-France.

Email addresses: `claisse@ann.jussieu.fr` (A. Claisse), `frey@ann.jussieu.fr` (P. Frey)

a data format more suitable for further storage, exchange, computation or visualization purposes.

More specifically, we consider the following initial value Hamilton-Jacobi problem, posed in a computational domain $\mathcal{D} \subset \mathbb{R}^3$, endowed with initial and boundary conditions.

Find $u(t, x) \in C_t^1 \times C_x^2$, such that :

$$\begin{cases} \frac{\partial u}{\partial t}(t, x) = |\nabla u(t, x)|(\kappa(u)(t, x) + \alpha d(x)), & \forall x \in \mathcal{D} \text{ and } \forall t \in \mathbb{R}^+, \\ u(0, \cdot) = u_0, \\ \langle u(t, x), n(t, x) \rangle = 0, & \forall x \in \partial\mathcal{D}, \end{cases} \quad (1)$$

where V denotes a point set, $d(x)$ is here the unsigned distance from any point in \mathbb{R}^3 to V , $\kappa(u)(t, x) = \nabla \cdot n(u)(t, x) = \left(\nabla \cdot \left(\frac{\nabla u}{|\nabla u|} \right) \right)(t, x)$ is the local mean curvature of the surface considered, hereafter denoted $\Gamma(t)$ and α is a real. Practically, the sign of α depends on the initial condition $u(0, x)$; *i.e.*, if $\Gamma(0)$ is enclosing all points in V , then we set $\alpha = 2$ otherwise, if $\Gamma(0)$ is inside the volume delimited by V then $\alpha = -2$. Note that the distance function d does not depend on the time variable since the set V remains unchanged during the computation. Hence, d will be computed only at the initial stage and will remain fixed at all times. Since d is bounded on the domain \mathcal{D} , we can take \mathcal{D} as the unit domain $[0, 1]^3$ thanks to a proper change of variable.

Furthermore, we consider the evolution of an implicitly defined regular surface $\Gamma(t) = \{x \in \mathcal{D}, u(t, x) = 0\}$. Then, we define the solution Γ as a specific occurrence of the surface $\Gamma(t)$ satisfying a convergence criterion defined hereafter. Within the level set context, the function u is classically defined to be negative inside the domain $\Omega(t)$ bounded by $\Gamma(t)$, positive outside $\Omega(t)$ and vanishing on $\Gamma(t)$ (*c.f.* Figure 1). The evolution of $\Gamma(t)$ is then governed by the nonlinear PDE (1) that imposes $\Gamma(t)$ to move in the normal direction at each point x with a velocity proportionnal to $\kappa(u)(t, x) + \alpha d(x)$.

Our level set equation modeling the geometric motion of Γ is expressed by a Hamilton-Jacobi type of equation, for all $(t, x) \in \mathbb{R}_+ \times \mathbb{R}^2$:

$$\frac{\partial u}{\partial t}(t, x) + v_n(t, x)|\nabla u(t, x)| = 0, \quad (2)$$

where $v_n(t, x) = \left\langle \frac{d\Gamma(t)}{dt}, n(u)(t, x) \right\rangle$ denotes the velocity field in the direction normal to the level set passing through x . In our case, $v_n(t, x) = \kappa(u)(t, x) + \alpha d(x)$.

We observe that our unsteady model relies on two terms. On the one hand, the fitting term αd involves the distance to the data set V and characterizes the *attraction* of the curve to V , whatever initial condition is considered. On the other hand, the term $\kappa(u)$ can be compared with a *surface tension* term and allows to control the regularity of $\Gamma(t)$ during its evolution (as κ depends on the normal $n(u)$). The solution Γ is obtained when all the points

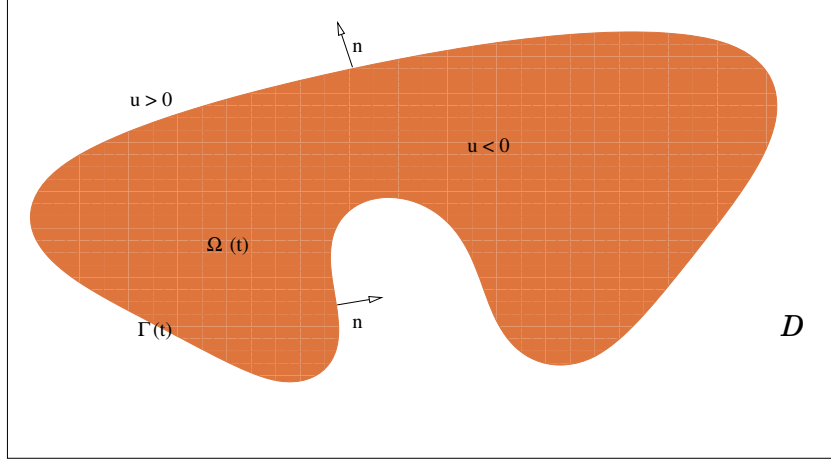


Figure 1: *Level set definitions.*

of V have a value u corresponding to, or close to, zero. This gives a straightforward test for checking the convergence of the numerical scheme, as will be seen later. Theoretically, the solution we are looking for is a viscosity solution that coincides with Γ .

Finally, the initial condition u_0 , that appears in (1), is a signed distance function to a non-intersecting, closed and orientable regular initial surface, chosen as simple as possible (a circle in dimension two, a sphere in dimension three, for instance). In other words, $u_0(x) = \pm d_0(x)$, where d_0 is the distance to $\Gamma(t = 0)$. The initial condition $u_0(x)$ is not subjected to any geometric requirement with respect to V , *i.e.*, the points of V can be inside or outside or on each side of the zero level set (cf. Section 5). Notice also that ∇u does not vanish on the set $\Gamma(t)$, and this will be useful for numerical results.

2. Minimization problem

The Cauchy problem (1) can be understood as the gradient flow equation associated to the minimization of a functional of the domain defined by the zero isovalue of the level function u . More precisely, let Ω_0 be a bounded reference domain of \mathbb{R}^d and let $\theta \in W^{1,\infty}(\mathbb{R}^d, \mathbb{R}^d)$ be a shape deformation, we denote by

$$\Omega_\theta = (I_d + \theta)(\Omega_0)$$

the considered variations of this domain. It is well-known that for θ sufficiently small, $(I_d + \theta)$ is a diffeomorphism in \mathbb{R}^d [27]. Given Ω_0 , we will call shape derivative of a functional $J(\Omega_\theta) = J((I_d + \theta)(\Omega_0))$, denoted by $J'(\Omega_0)(\theta)$, the Fréchet derivative of J at θ in $W^{1,\infty}(\mathbb{R}^d, \mathbb{R}^d)$ and the following result recalls the shape derivatives for some classical functionals.

Lemme 2.1. *Let Ω_0 be a smooth bounded domain of \mathbb{R}^d and $f \in W^{1,1}(\mathbb{R}^d)$ and $g \in W^{2,1}(\mathbb{R}^d)$ two functions independent of the domain variation and let define*

$$J_1(\Omega) = \int_{\Omega} f(x) dx, \quad \text{and} \quad J_2(\Omega) = \int_{\partial\Omega} g(x) ds(x).$$

Then, for any $\theta \in W^{1,\infty}(\mathbb{R}^d, \mathbb{R}^d)$,

$$\begin{aligned} J'_1(\Omega_0)(\theta) &= \int_{\partial\Omega_0} \theta(x) \cdot n(x) f(x) \, ds \\ J'_2(\Omega_0)(\theta) &= \int_{\partial\Omega_0} \theta(x) \cdot n(x) \left(\frac{\partial g}{\partial n} + \kappa g \right) (x) \, ds \end{aligned}$$

where n is the unit outer normal and κ is the mean curvature of Ω_0

Now, we can introduce the following functional:

$$J(\Omega) = \alpha \int_{\Omega} d(x) \, dx + \int_{\partial\Omega} 1 \, ds, \quad (3)$$

which has the form $J(\Omega) = J_1(\Omega) + J_2(\Omega)$, with $f(x) = \alpha d(x)$ and $g(x) = 1$. Notice that this functional is composed of two terms (see previous section for the analogy): the first term will drive the considered domain to the point set, where $d(x) = 0$ (the *attraction*) and the second term will enforce some regularity (the *surface tension*) for its boundary. We can assert that for $\theta \in W^{1,\infty}(\mathbb{R}^d, \mathbb{R}^d)$,

$$J'(\Omega_0)(\theta) = \int_{\partial\Omega_0} \theta(x) \cdot n(x) (\alpha d(x) + \kappa(x)) \, ds,$$

which means that minimizing the functional J amounts to driving its boundary through a motion with speed $v(x) = -(\alpha d(x) + \kappa(x)) \cdot n(x)$. This motion can be carried out within a level set formulation, then $u(t, x)$ is the level set function used to represent the domain $\Omega(t)$. As expected, the corresponding Hamilton-Jacobi equation is exactly Equation (1):

$$\frac{\partial u}{\partial t}(t, x) = |\nabla u(t, x)|(\kappa(u)(t, x) + \alpha d(x)), \quad \forall x \in \mathcal{D} \text{ and } \forall t \in \mathbb{R}^+.$$

Remarks:

1. Note that this scheme will hopefully drive the domain to a *local* minimum of functional J . Indeed, functional J admits a global minimum corresponding to a single point when J vanishes. This reinforce the need to rely on a stopping criterion at the numerical level.
2. Furthermore, it is well-know [2] that shape optimization problems can be very difficult to implement numerically. The reason being that they do not necessarily admit minima within the set of considered domains.

3. Related works

There are several classes of approaches for reconstructing a surface from a point set, that can be schematically classified as follows.

- *Geometrical approaches*

Computational geometry approaches are mainly based on Delaunay triangulations or dual Voronoï diagrams [19, 26, 16]. The desired surface is then extracted from a triangulation of the convex hull of the data set V . However, this surface is only piecewise affine and not C^2 continuous as desired. Moreover, such approaches cannot deal efficiently with noisy data usually.

- *Volumetric approaches*

Assuming that a signed distance function d to the points of the data set V has been defined on a Cartesian grid, most volumetric approaches construct a piecewise affine discretization of the zero level set of d using efficient and straightforward heuristic algorithms [6, 7, 28], *e.g.* Marching polyhedra [9, 32], based on pre-defined patterns. Nevertheless, the reconstructed manifold may be non orientable and this drawback is known to be one of the most critical problems in the reconstruction process.

- *Snake contour approaches*

Typically, this data extraction method, designed for processing images, relies on the displacement of points of an active contour towards areas with high gradients of the intensity function (associated with the pixels), keeping the contour characteristics like curvature or points distribution [11, 31]. The algorithm consists in contracting or dilatating the initial contour using a sum of internal (related to the shape of the contour) and external (related to the location of the contour) energies. However, a large number of points may be necessary to resolve the high local curvature areas that are difficult to identify *a priori*.

- *Continuous PDE-driven approaches*

Methods based on the resolution of partial differential equations (PDE), and especially those using the level set formalism introduced by [37, 44], are efficient to handle dynamically moving interfaces. In such approach, the deformation process, governed by an evolutionary PDE, starts from an initial closed surface, enclosing all the points of V . This contour is deformed, at each point, in the normal direction, with a value proportional to the local curvature, until it matches the sample. This continuous framework requires nevertheless the tedious estimation of discrete derivatives on the spatial discretization of the domain \mathcal{D} .

We consider the reconstruction problem on the continuous level, by developing a continuous model based on the resolution of a nonlinear PDE, like in [33, 44]. The formalism of the level set method (coupling an implicit function definition and its evolution in time) is a convenient way to link the construction of this surface and its representation. However, instead of solving classically the PDE on structured grids, we introduce here adapted anisotropic

triangulations containing highly stretched elements in the vicinity of the data set. As will be shown (Section 5), this feature considerably improves the numerical approximation of the manifold and limits the dissipative effects of the numerical schemes.

To summarize, the main features of our reconstruction method are related to:

1. the definition of the nonlinear PDE model and the level set formalism of this problem,
2. the numerical schemes used to solve the PDE,
3. the creation of anisotropic adapted triangulations,

and will be analyzed in the following sections. In addition, several application examples will be given to show the efficiency of this approach.

4. Existence and uniqueness of a solution

The notion of viscosity solution has been introduced in the 80's by [17, 18] to give a physical meaning to the solutions of Hamilton-Jacobi equations. The viscosity theory is usually employed to analyse PDEs of the form:

$$F(x, u, Du, D^2u) = 0 \quad \text{in an open set } \mathcal{D} \in \mathbb{R}^3, \quad (4)$$

where Du denotes the gradient of the unknown function u , D^2u is the Hessian matrix of u and F is defined by $F : \mathbb{R}^3 \times \mathbb{R} \times \mathbb{R}^3 \times \mathcal{S}_3(\mathbb{R}) \mapsto \mathbb{R}$, where $\mathcal{S}_3(\mathbb{R})$ is the set of square symmetric scalar 3×3 matrices.

The fundamental idea of the theory is that under the viscosity solution concept, the solution u needs not to be differentiable everywhere. It provides an equivalent definition for the classical notions of supersolution and subsolution of an equation posed in \mathbb{R}^3 , that does not involve the derivatives of the function, but use test functions.

Generally, viscosity solutions are considered as weak solutions that allow in particular to extend the maximum principle properties to less regular solutions. Here Equation (1) belongs to the wider class of mean curvature flow equations that have been thoroughly studied. Hence, the fundamental results obtained by [24, 12] can be applied on our equation, without restriction. In order for this theory to be used, the function F must comply with the following necessary conditions:

- a monotonicity condition:

$$F(x, r, p, X) \leq F(x, s, p, X) \quad \text{with} \quad r \leq s, \quad (5)$$

where $r, s \in \mathbb{R}$ and $x, p \in \mathbb{R}^3$,

- an ellipticity condition:

$$F(x, r, p, Y) \leq F(x, r, p, X) \quad \text{with} \quad Y \geq X, \quad (6)$$

where $X, Y \in \mathcal{S}_3(\mathbb{R})$ endowed with the usual order relationship.

We consider the following function F corresponding to the steady part of Problem (1) (allowing us to omit the time variable t):

$$F(x, u, \nabla u, \mathcal{H}(u)) = -|\nabla u(x)| \left(\alpha d(x) - \left(\nabla \cdot \left(\frac{\nabla u}{|\nabla u|} \right) \right) (x) \right), \quad (7)$$

where $\mathcal{H}(u)$ is here the Hessian matrix of u .

Proposition 4.1. *Function F defined by (7) satisfies the two previous necessary conditions (5) and (6).*

Proof. We denote by $r = u$, $p = \nabla u$ and $X = \mathcal{H}$. The function F can be then rewritten as:

$$F(x, r, p, X) = -d(x)\text{Tr}(X) + \text{Tr} \left(\frac{p \otimes p}{|p|^2} X \right) - \alpha d(x)|p|,$$

where, in dimension two,

$$\text{for } p = \begin{pmatrix} p_1 \\ p_2 \end{pmatrix} \in \mathbb{R}^2, \quad \frac{p \otimes p}{|p|^2} = \frac{1}{|p|^2} \begin{pmatrix} p_1^2 & p_1 p_2 \\ p_1 p_2 & p_2^2 \end{pmatrix}.$$

We observe that F does not depend directly on r , *i.e.* $F(x, r, p, X) = F(x, p, X)$, and thus the monotonicity condition (5) is verified.

To establish the condition (6), we denote by $A(p) = I - \frac{p \otimes p}{|p|^2}$. Then, for $p \neq 0$, we have:

$$A(p) = \begin{pmatrix} \frac{p_1^2}{|p|^2} & \frac{-p_1 p_2}{|p|^2} \\ \frac{-p_1 p_2}{|p|^2} & \frac{p_2^2}{|p|^2} \end{pmatrix}.$$

Matrix $A(p)$ is positive and semi-definite. Indeed, posing $v = \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} \in \mathbb{R}^2 - \{0\}$, yields

$$v^T A(p) v = \frac{1}{|p|^2} (v_1 p_2 - v_2 p_1)^2 \geq 0.$$

Furthermore, matrix $A(p)$ is symmetric, his eigenvalues are non negative and there exists an orthonormal basis for the decomposition $D = P^T A(p) P$ with P (resp. D) an orthogonal (resp. diagonal) matrix with positive values. Writing $A(p) = \sigma \sigma^T$, with $\sigma = P D^{1/2}$ allows us to obtain:

$$\text{Tr}(A(p)X) = \text{Tr}(\sigma \sigma^T X) = (\sigma^T X \sigma) = \sum_{i=1}^2 \sigma_i^T X \sigma_i,$$

where σ_i is the i^{th} column of σ .

Assuming that for $X, Y \in \mathcal{S}_2(\mathbb{R})$, $Y \geq X$, we have:

$$\forall i \in \{1, 2\}, \quad \sigma_i^T X \sigma_i \leq \sigma_i^T Y \sigma_i, \quad \text{and} \quad \frac{\sigma_i^T X \sigma_i}{|p|^2} \leq \frac{\sigma_i^T Y \sigma_i}{|p|^2}.$$

Since $d \geq 0$, the function F is thus elliptically degenerated for $p \neq 0$ and this completes the proof. \square

We consider now the associated parabolic problem of (4):

$$\frac{\partial u}{\partial t} + F(t, x, u, \nabla u, \mathcal{H}(u)) = 0.$$

The existence and the uniqueness of the solution of Problem (1) comes from the works of [12] and [24] where the function F has to be $C^0([0, T] \times \overline{\mathcal{D}} \times \mathbb{R} \times (\mathbb{R}^3 - \{0\}) \times \mathcal{S}_3(\mathbb{R}))$.

Notice here that u_0 must be C^2 continuous, however, there is no explicit requirement on the regularity of \mathcal{D} and of $\partial\mathcal{D}$.

We introduce now the main following result regarding the viscosity solution.

Theorem 4.2. *We consider the following initial boundary value problem:*

$$\begin{cases} u_t + F(t, x, u, \nabla u, \mathcal{H}(u)) = 0, & \text{on }]0, T[\times \mathcal{D}, \\ \langle u(x), n(u)(x) \rangle = 0, & \text{on } \partial\mathcal{D}, \end{cases} \quad (8)$$

with $u(0, x) = u_0(x)$ for $x \in \overline{\mathcal{D}}$.

Assuming conditions (5) and (6) are satisfied, then, for any function $u_0 \in C^0(\overline{\mathcal{D}})$, there exists a unique viscosity solution $u \in C^0([0, T] \times \overline{\mathcal{D}})$ such that $u(0, x) = u_0(x)$.

Classically, in such PDE problem, the analytical solution u cannot be exhibited and thus a numerical approximation u_h of this solution is sought. To this end, we propose hereafter a semi-implicit numerical scheme to solve numerically Problem (1).

5. A semi-implicit numerical scheme

In most problems that require the discretization of the computational domain, the accuracy of the solution and its regularity are strongly related to the quality of the mesh, as well as to its adaptation to the problem at hand. With the type of discontinuities occurring with level set methods, standard centered finite difference schemes for computing the curvature term tend to become inaccurate and unstable, *i.e.*, lead to spikes in the curvature error after several iteration steps. Refining locally the Cartesian mesh may only postpone the problem without totally eliminating it [34]. Furthermore, the mesh cannot be refined indefinitely to capture vanishing gradients as it would then lead to restrictive time steps with explicit schemes. On the other hand, the finite element method has become eminently popular in engineering applications as it involves a variational formulation of the problem and looks for solutions in suitable functional spaces. It offers the flexibility of dealing with an unstructured triangular mesh of the domain, possibly adapted to the geometry of the domain boundary as well as to the solution variation. However, the approximation of second order derivatives is related to the degree of the polynomials associated with the shape functions. Obtaining an accurate nodal value of the second order terms in Problem (1) would require using at least third degree polynomials, thus leading to a substantial increase of the number of degrees of freedom in our problem. In addition, essentially non oscillatory (ENO) schemes

are sometimes relatively tedious to implement on triangulations (see [1] for a discussion and details of the implementation).

Finally, as pointed out by [34], using a local level set function is more robust in calculating the curvature than directly differentiating an interpolating function like a spline for instance. Next, we propose an alternative for computing a curvature value at each mesh vertex without having to resort on high order interpolation schemes. Moreover, this approach is easy to implement on unstructured meshes.

From now, we consider an unstructured simplicial triangulation T_h of the domain \mathcal{D} containing N_n nodes. We assume that the value of the distance function $d(x_i)$ between the points of V and any vertex $x_i \in T_h$ is known, *i.e.*, has been previously computed.

5.1. Approximation of the time derivative

In Problem (1), the first order differential operator with respect to the time variable t can be discretized using a classical Euler finite difference scheme with a time step Δt :

$$\frac{\partial u}{\partial t} \sim \frac{u_i^{n+1} - u_i^n}{\Delta t},$$

where $u_i^n = u(t^n, x_i)$, $t^n = n \Delta t$ and x_i is a vertex of T_h and we denote $d_i = d(x_i)$. For efficiency purposes, we propose the following semi-implicit scheme for solving Problem (1):

$$u_i^{n+1} - |\nabla u^n|_i \Delta t \left(\nabla \cdot \left(\frac{\nabla u^{n+1}}{|\nabla u^n|} \right) \right)_i = |\nabla u^n|_i d_i \Delta t \alpha + u_i^n, \quad \begin{array}{l} \text{for all } t \in [0, T], \\ \text{for all } i \in \{1, \dots, N_n\}. \end{array} \quad (9)$$

The term $|\nabla u^n|$ on the denominator appeals the following comment. A renormalization procedure is applied during the numerical resolution, to ensure that the level set function u remains close to a distance function. In other words, the modulus of the gradient value of u is always bounded by above by some constant which is related to the unit domain size. Since we assume the solution u is regular (typically C^m , $m \geq 1$), then ∇u is C^{m-1} and at least continuous, and furthermore $\nabla u(t, x) \neq 0$ for $x \in \Gamma(t)$. Hence, to avoid any numerical problem, we also bound the modulus of the gradient term on the denominator by a minimal value ε .

5.2. Stability issue

In this sub-Section, we prove the stability of the time discretization of our Problem (1). We denote by $k = |\nabla u^n|$ such that $k \neq 0$, then we have :

$$u^{n+1} - k \Delta t \left(\nabla \cdot \left(\frac{\nabla u^{n+1}}{k} \right) \right) = u^n + k \Delta t \alpha d, \quad \text{with} \quad \frac{\partial u^{n+1}}{\partial n} = 0 \text{ on } \partial \mathcal{D}.$$

Let $Au = u - k \Delta t \left(\nabla \cdot \left(\frac{\nabla u}{k} \right) \right)$. Notice that A is not a self-adjoint operator for the L^2 scalar product, but for $\langle u, v \rangle = \int \frac{uv}{k}$, since we have:

$$\begin{aligned} \langle Au, v \rangle &= \langle u, v \rangle - \Delta t \int k \left(\nabla \cdot \left(\frac{\nabla u}{k} \right) \right) \frac{v}{k} \\ &= \langle u, v \rangle + \Delta t \int \frac{\langle \nabla u, \nabla v \rangle}{k} > 0. \end{aligned}$$

In particular, we have $\langle Au, u \rangle \geq \|u\|^2$, then let $\lambda \in \sigma(A)$ be an eigenvalue, we have $\lambda \geq 1$. Indeed, A is diagonalisable (since A is a self-adjoint operator with compact inverse) and $\sigma(A) = \{\lambda_0, \lambda_1, \dots\}$ with $\lambda_0 \leq \lambda_1 \leq \dots$, $\lambda_i \xrightarrow{i \rightarrow \infty} \infty$. Moreover, $\forall i \in \mathbb{N}, \exists u_i \in L^2(\mathcal{D}, \frac{dx}{k})$, such that $Au_i = \lambda_i u_i$.

Our time discretization of the Problem (1) is then unconditionally stable on the time step, provided $k \neq 0$.

5.3. Approximation of the spatial derivatives

In order to ensure the robustness, and to a lesser extent, the efficiency of the numerical method, we need to compute the first (gradient) and the second (mean curvature) order derivatives of the level set function u as accurately as possible. With classical finite difference/element approximations, the level set function u is defined at the nodal points, and so are the spatial derivatives. Hence, we would like to define these derivatives at the vertices of the triangulation T_h . Next, we propose a method for evaluating the gradient and the mean curvature at each vertex of T_h , based on an L^2 projection.

As usual, for a given time t^n , we denote hereafter the solution $u_i^n = u_i$.

5.3.1. Gradient approximation

Introducing the barycentric coordinates ω_{Ki_2} of a vertex x_{i_2} at each triangle $K \in T_h$ leads to write:

$$u|_K = \sum_{x_{i_2} \in K} u_{i_2} \omega_{Ki_2}, \quad \text{and} \quad (\nabla u)|_K = \sum_{x_{i_2} \in K} u_{i_2} \nabla \omega_{Ki_2}. \quad (10)$$

Then, for each vertex x_i in the triangulation T_h , the discrete gradient is defined as the L^2 projection using the operator Π_h :

$$(\nabla u)_i = (\Pi_h(\nabla u_h))_i = \frac{\sum_{K \in \mathcal{B}_i} |K| (\nabla u)|_K}{|\mathcal{B}_i|} = \frac{\sum_{K \in \mathcal{B}_i} |K| \sum_{x_{i_2} \in K} u_{i_2} \nabla \omega_{Ki_2}}{|\mathcal{B}_i|}, \quad (11)$$

where $|K|$ is the area of triangle K , \mathcal{B}_i is the support of x_i , *i.e.*, the set of triangles containing vertex x_i and $|\mathcal{B}_i| = \sum_{K \in \mathcal{B}_i} |K|$.

5.3.2. Mean curvature approximation

Once the gradient approximation has been obtained at each mesh vertex, the same procedure can be applied to each component of the gradient vector to supply a mean curvature value at each vertex as well. The previous formulas allows us to write:

$$(\nabla(\nabla u))|_K = \sum_{x_i \in K} (\nabla u)_i (\nabla \omega_{Ki})^T,$$

where we considered the row vector $(\nabla \omega_{Ki})^T$, in order to calculate the product. With this definition, $(\nabla(\nabla u))|_K$ is a square matrix. The aim is to obtain the value $(\nabla(\nabla u))_i$ at each point x_i . Hence, using the previous formulas (10) and (11), we compute successively the coefficients of the Hessian matrix:

$$(\nabla(\nabla u))_i = \frac{\sum_{K \in \mathcal{B}_i} |K| \nabla(\nabla u)|_K}{|\mathcal{B}_i|} = \frac{\sum_{K \in \mathcal{B}_i} |K| \left(\sum_{x_{i_2} \in K} (\Pi_h(\nabla u_h))_{i_2} \right) (\nabla \omega_{Ki_2})^T}{|\mathcal{B}_i|}, \quad (12)$$

and the trace of the Hessian matrix:

$$(\nabla \cdot (\nabla u))_i = \frac{\sum_{K \in \mathcal{B}_i} |K| \left(\sum_{x_{i_2} \in K} \left(\frac{\sum_{L \in \mathcal{B}_{i_2}} |L| \left(\sum_{x_{i_3} \in L} u_{i_3} \langle \nabla \omega_{Li_3}, \nabla \omega_{Ki_2} \rangle \right)}{|\mathcal{B}_{i_2}|} \right) \right)}{|\mathcal{B}_i|}, \quad (13)$$

where $\langle \cdot, \cdot \rangle$ is the Euclidian scalar product. Futhermore, the local mean curvature at a mesh vertex x_i is then defined as:

$$\kappa_i = \left(\nabla \cdot \left(\frac{\nabla u}{|\nabla u|} \right) \right)_i = \frac{\sum_{K \in \mathcal{B}_i} |K| \left(\sum_{x_{i_2} \in K} \left(\frac{\sum_{L \in \mathcal{B}_{i_2}} |L| \left(\sum_{x_{i_3} \in L} u_{i_3} \langle \nabla \omega_{Li_3}, \nabla \omega_{Ki_2} \rangle \right)}{|\nabla u|_{i_2} |\mathcal{B}_{i_2}|} \right) \right)}{|\mathcal{B}_i|}. \quad (14)$$

From the numerical point of view, these two approximations are relatively easy to implement. The sole difficulty is related to the fast identification of all triangles in the sets \mathcal{B}_i , for all i . This can be achieved using appropriate data structures, especially since T_h is kept unchanged during all the evolution of the curve [25], as will be seen in the Section (5.1).

5.4. Consistency

For the sake of simplicity, we consider a two dimensional Cartesian grid T_h of generic size h and we denote K a square cell of T_h , $x_{i,j}$ being the point at which we calculate the evaluation of the curvature. Let $\mathcal{B}_{i,j}$ be the ball of the point $x_{i,j}$, *i.e.*, the set of the cells containing $x_{i,j}$. By analogy with the formula (14), we fix $a = |\nabla u|$ for all u and then, the curvature evaluation yields:

$$\left(\nabla \cdot \left(\frac{\nabla u}{a} \right) \right)_{i,j} = \frac{1}{4} \frac{\sum_{K \in \mathcal{B}_{i,j}} |K| \left(\sum_{x_{i_2,j_2} \in K} \left(\frac{\sum_{L \in \mathcal{B}_{i_2,j_2}} |L| \left(\sum_{x_{i_3,j_3} \in L} u_{i_3,j_3} \langle \nabla \omega_{L,i_3,j_3}, \nabla \omega_{K,i_2,j_2} \rangle \right) \right)}{a_{i_2,j_2} |\mathcal{B}_{i_2}|} \right)}{|\mathcal{B}_i|}. \quad (15)$$

Here, the coefficient $\frac{1}{4}$ comes from the calculation of the term $\langle \nabla \omega, \nabla \omega \rangle$ in square cells. At each grid point $x_{i,j}$, we can write the following equation:

$$\left(\nabla \cdot \left(\frac{\nabla u}{a} \right) \right)_{i,j} = \frac{1}{64} \sum_{K \in \mathcal{B}_{i,j}} \sum_{x_{i_2,j_2} \in K} \left(\frac{\sum_{L \in \mathcal{B}_{i_2,j_2}} \sum_{x_{i_3,j_3} \in L} u_{i_3,j_3} \langle \nabla \omega_{L,i_3,j_3}, \nabla \omega_{K,i_2,j_2} \rangle}{a_{i_2,j_2}} \right).$$

The number of cells in $\mathcal{B}_{i,j} \times$ the number of squares in \mathcal{B}_{i_2,j_2} appears in the denominator.

The coefficients contributing to the value of the stencil of $\left(\nabla \cdot \left(\frac{\nabla u}{a} \right) \right)_{i,j}$ can be found in Appendix A.

Proposition 5.1. *On Cartesian grids, the discrete operator $\left(\nabla \cdot \left(\frac{\nabla}{|\nabla|} \right) \right)_{i,j}$ is consistent with the continuous operator $\nabla \cdot \left(\frac{\nabla}{|\nabla|} \right)$.*

The proof of this proposition is detailed in Appendix B.

5.5. Matrix form of the scheme

For all u^n , we fix $|\nabla u^n|$ and by defining the discretization matrix M by:

$$u \mapsto Mu = \nabla \cdot \left(\frac{\nabla u}{|\nabla u^n|} \right),$$

and by introducing D the matrix of diagonal terms $|\nabla u^n|_i$, Scheme (9) becomes:

$$\begin{aligned} ((I - \Delta t DM)u)_i^{n+1} &= u_i^n + \Delta t |\nabla u^n|_i d_i \alpha. \end{aligned} \quad \begin{array}{l} \text{for all } t \in [0, T], \\ \text{for all } i \in \{1, \dots, N_n\}. \end{array} \quad (16)$$

This semi-implicit scheme (16) requires the resolution of a linear system. This relies on the following proposition.

Proposition 5.2. *The matrix $(I - \Delta t DM)$ is invertible.*

The proof of this Proposition is detailed in Appendix C.

6. Application to curve and surface reconstruction problems

In this Section, we show that the initial boundary value Problem (1) is especially well-suited for dealing with surface reconstruction problem. This problem is of interest in various fields of applications, for instance in numerical simulations, in biomedical, reverse engineering, and visualization. In such cases, the initial data (point set) may not be always well defined and may contain noise, for instance. Rather than searching for an interpolating surface, we would prefer to find a surface that fits at best the set of points, thus overcoming the noise problem, corresponding to a mean regular surface sufficiently close to the point set. We first provide some informations about the numerical aspects of our approach. Then, we give several examples of curve and surface reconstructions.

Schematically, Problem (1) is solved in three stages as shown in Figure 2. A preliminary stage consists in computing the unsigned distance function on a regular triangulation covering a large convex area \mathcal{D} encompassing all data points of the set V . Next, a triangulation locally adapted to the zero level of the unsigned distance function is generated, to provide a minimal element size close to the data set and thus allowing for a better accuracy in this neighborhood. The unsigned distance function is then recomputed at the vertices of this adapted triangulation as well as the initial condition u_0 , that will be used to solved the problem. The numerical resolution consists in a main time loop, corresponding to the discretization of the time derivative of u , in which the stiffness matrix is updated and the resulting linear system is solved using an iterative algorithm. We will now briefly describe these stages and provide numerical information about the implementation.

6.1. Definition of the distance function

The unsigned distance to an arbitrary set of points V is the function $\mathcal{D} \ni x \mapsto d(x)$ defined by:

$$d(x) = \inf_{y \in V} \|y - x\| = d(x, V),$$

where $\|\cdot\|$ denotes here the Euclidean norm in \mathbb{R}^d . Since this Euclidean distance function is 1-lipschitz continuous, *i.e.*,

$$\forall x, y \in \mathcal{D} \quad |d(x, V) - d(y, V)| \leq \|x - y\|,$$

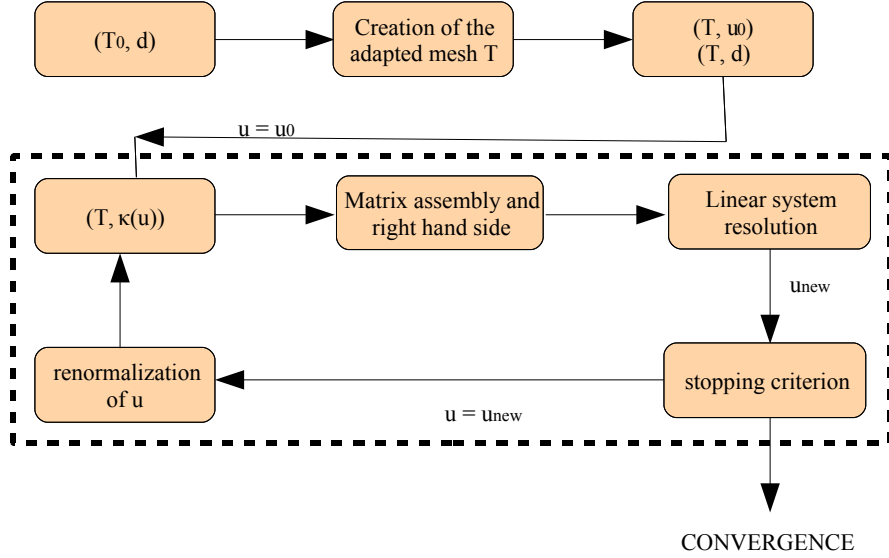


Figure 2: *General scheme for solving Problem (1).*

Rademacher's theorem allows to conclude that this distance function d is differentiable almost everywhere on \mathcal{D} . In particular, d solves the Eikonal equation almost everywhere:

$$|\nabla d(x)| = 1, \quad a.e. \ x \in \mathcal{D},$$

and is such that $d(x) = 0$ for $x \in V$. Actually, the distance function d to V is a (non unique) viscosity solution of this Eikonal equation.

From the practical point of view, we compute the distance function to V at the vertices of a triangulation T covering \mathcal{D} using a greedy algorithm of complexity $\mathcal{O}(n \times |V|)$, where n denotes the number of vertices in T and $|V|$ the number of points in V . We are aware that other methods exist to compute the unsigned distance function to a point set more efficiently (for example considering the Eikonal equation as the steady state of an evolution equation). But as will be seen, the number of points n in an adapted anisotropic triangulation T is usually small and this computation is only performed twice during the initialization stage (on a regular triangulation and on an adapted triangulation).

6.2. Anisotropic mesh adaptation

Here, we assume the unsigned distance function d has been computed at the vertices of a regular triangulation T of \mathcal{D} . Actually, the zero level set $\{x \in T, d(x) = 0\}$ is a crude approximation of the distance to the set of points V and is highly dependent of the initial regular triangulation T on which it is computed and of the sampling of V . Since d is used to solve the evolution Equation (1) at each time steps, we would like to improve its numerical evaluation. This can be achieved by adapting the triangulation T to our purpose using Riemannian metrics [25]. Anisotropic adaptation allows to control the size, the shape and

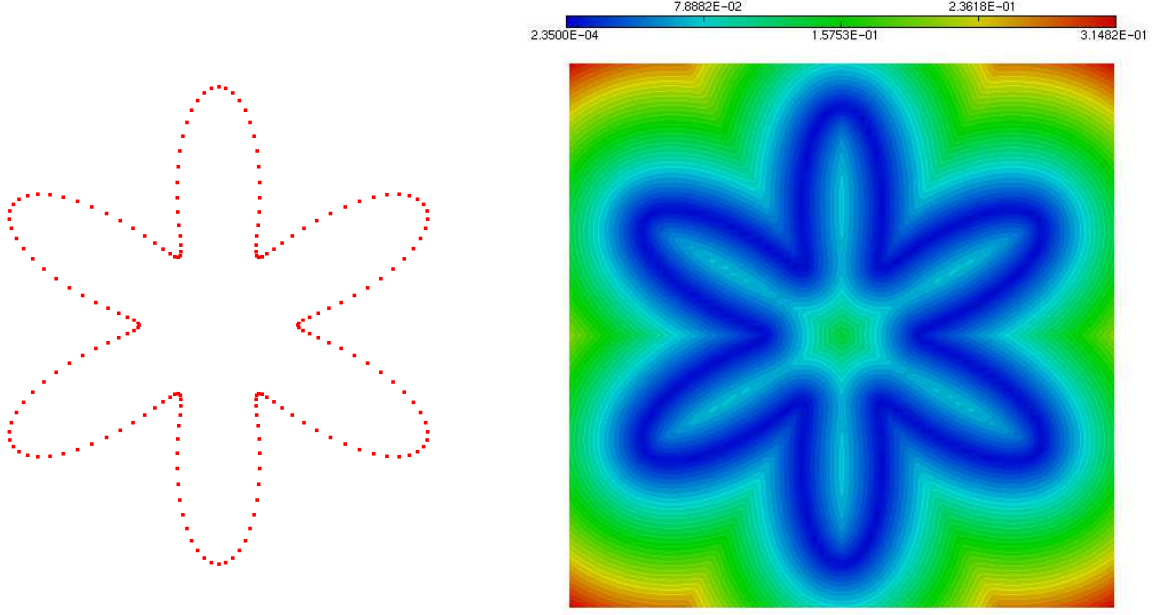


Figure 3: *Data set V (left) and unsigned distance function related to this set (right).*

the stretching to mesh elements, as compared to isotropic adaptation that controls only the size. For example, on Figure 4, the isotropic triangulation (left) contains almost four times more vertices than the anisotropic triangulation (right), for the same level of accuracy. We briefly recall this notion and its application to mesh adaptation.

Consider two orthogonal unit vectors e_1 and e_2 in \mathbb{R}^2 and suppose we want to force a triangle $K \in T$ to be aligned with the direction e_1 and to be very flat along e_2 . This can be obtained by prescribing a metric \mathcal{M} as:

$$\mathcal{M} = P \begin{pmatrix} \frac{1}{l_1^2} & 0 \\ 0 & \frac{1}{l_2^2} \end{pmatrix} P^t,$$

where P is the transformation matrix from the canonical basis to the basis (e_1, e_2) , l_1 (resp. l_2) is the desired length along e_1 (resp. e_2).

To compute an accurate discretization of the zero level set of the approximation of u solving Problem (1), we need to increase the density of points in the vicinity of V . By prescribing the following metric $\mathcal{M}(x)$ at all vertices of T , we are able to control approximation error (and the Hausdorff distance) between the continuous and the discrete solution [21]:

$$\mathcal{M}(x) = P(x) \begin{pmatrix} \min \left(\max \left(\frac{c}{\epsilon} |\kappa(x)|, \frac{1}{h_{max}^2} \right), \frac{1}{h_{min}^2} \right) & 0 \\ 0 & \frac{1}{h_{max}^2} \end{pmatrix} P^t(x),$$

where h_{max} and h_{min} represent the stretchings of the mesh element, c is a constant depending on the space dimension and ϵ is a tolerance value on the Hausdorff distance between the

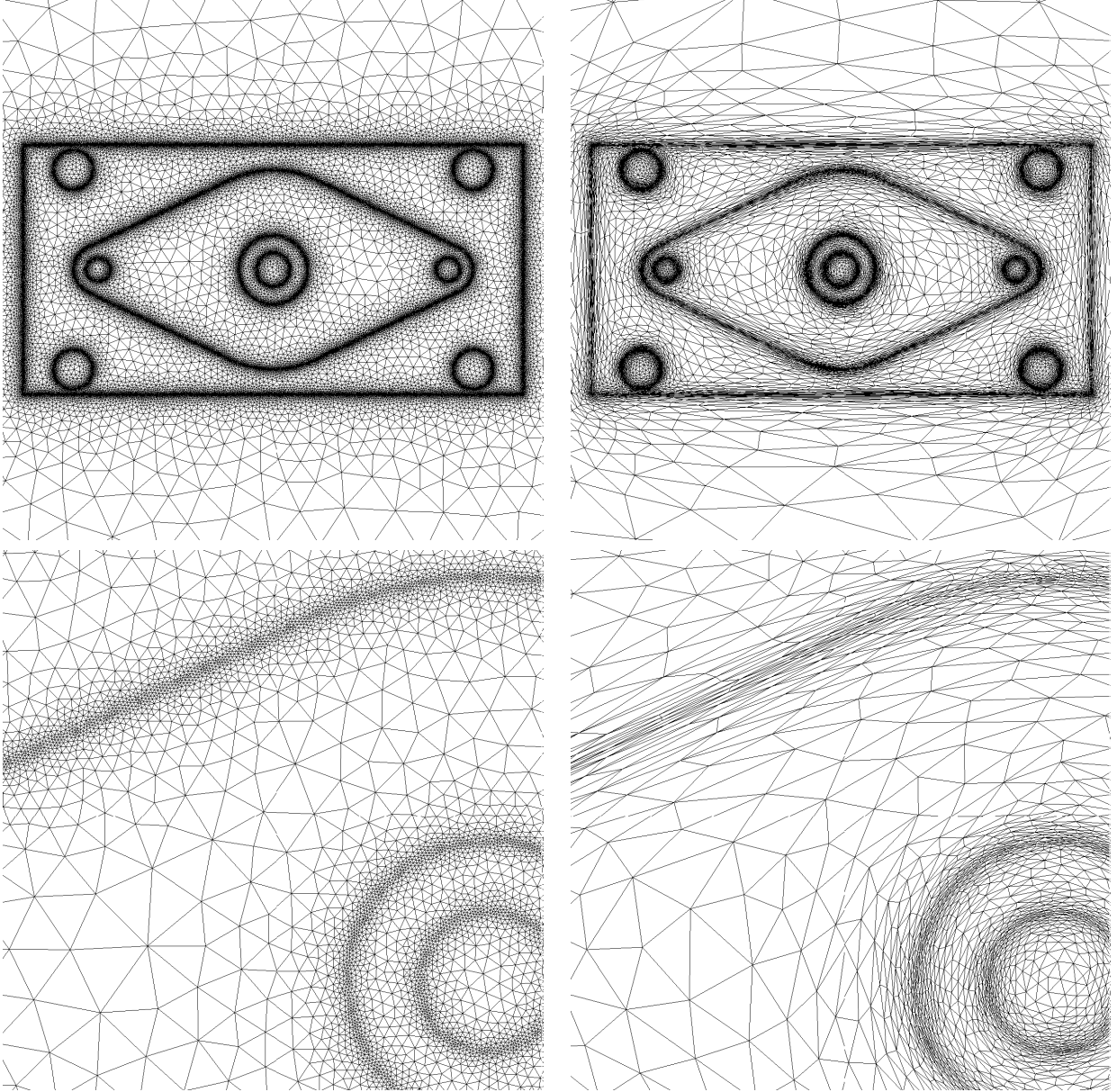


Figure 4: *Example of isotropic (left) and anisotropic (right) mesh adaptation based on an unsigned distance function defined at the mesh vertices. The adapted isotropic (resp. anisotropic) mesh contains 37,750 vertices (resp. 9885) corresponding to a minimal size of 0.001 unit (\mathcal{D} is the unit square).*

zero isovalue and its discretization [15, 21]. Such metric indicates that the element size is proportional to the local mean curvature $\kappa(x)$ of d along the tangent direction. Numerically, to compute the mean curvature term $\kappa(x)$ at the mesh vertices we can use the same scheme as we use to compute the first and second order derivatives of u , thanks to the Eikonal property of the function d (Section 5.2).

The regular triangulation is then adapted to the metric field \mathcal{M} defined at the mesh vertices by using classical mesh modification operations: edge collapsing, edge flipping, Delaunay point insertion and vertex relocation (see [25], for more details). Notice that the adapted triangulation will remain unchanged during the numerical resolution of Problem (1). The Figure 5 (left) illustrates the local mesh density generated in the vicinity of the zero level set of the unsigned distance function d . Furthermore, it is absolutely not required for the points of the data set V to be vertices of the adapted triangulation T . The latter have been created and inserted in T to fit at best the metric prescription \mathcal{M} , without considering V at this stage. Nevertheless, if a mesh vertex appears to be excessively close to a point of V , the vertex coordinates can be updated to match the point coordinates (if the triangulation remains valid). In addition, the anisotropic triangulation T has also revealed well suited for extracting an accurate piecewise polygonal approximation of the zero level set of the solution u solving Problem (1), for instance for visualization purpose (Figure 5, right).

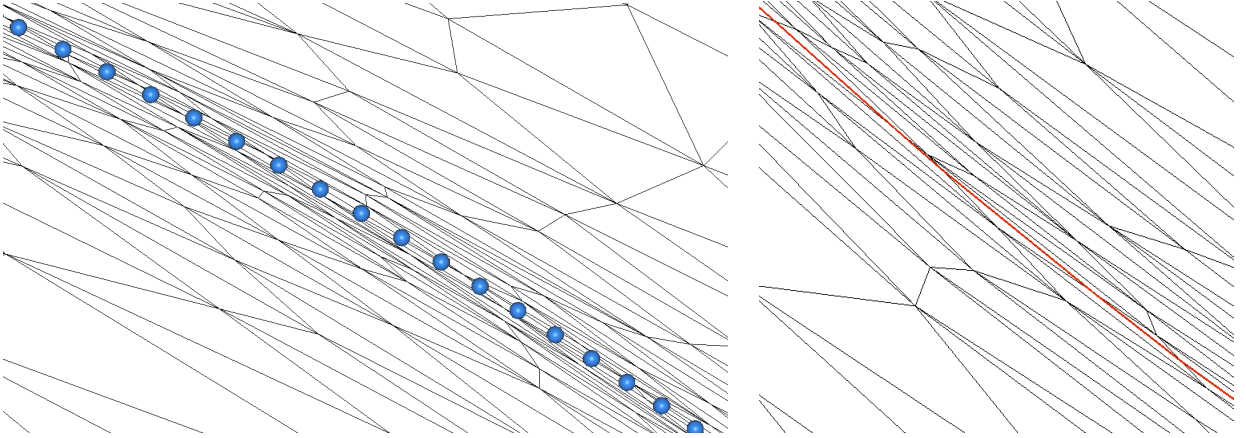


Figure 5: *Example of an anisotropically adapted triangulation T in the vicinity of the points of V (left). Local enlargement in the vicinity of the zero level set of the solution on the anisotropic triangulation T . The red line corresponds to a piecewise affine discretization of this level set (right).*

6.3. Numerical resolution

The initial condition u_0 and the unsigned distance function d are evaluated on the final adapted triangulation T . As described Figure 2, the numerical scheme consists in a main time loop in which, at each iteration, we compute the local mean curvature term $\kappa(u)$ at the mesh vertices, we update the stiffness matrix and the right hand side, and then we solve the resulting linear system using a GMRES algorithm. From numerical experiments, we have

seen the efficiency of introducing the following stopping criterion on the solution u_{new} :

$$\max_{x \in V} |\Pi_h u_{new}(x)| \leq \varepsilon,$$

corresponding to a L^2 projection of u_{new} on the points of V , for a small ε . In practice, this small value is expected to be related to the minimal mesh element size, usually taken as the minimal inradius of the element, as can be seen in the numerical examples section. It is well known that order one schemes lead to a numerical dissipation after a few iterations, *i.e.* the Eikonal property $|\nabla u| = 1$ is no longer satisfied. To overcome this problem, a classical renormalization stage is applied every few iterations [42].

We present now several numerical examples to illustrate the efficiency of our curve and surface reconstruction approach.

6.4. Curve reconstruction

In this section, we present several application examples of curve reconstruction. In these cases, we have set the parameter α in Equation (1) to the value 2 to achieve a good balance between the attraction term and the surface tension term. For these examples, we specify the mesh features (number of points, triangles and the minimal size), as well as the L^2 -error at convergence.

6.4.1. Uniform set of points

In this first example, the objective is to capture the boundary of a closed domain defined from a segmented binary image. In this example the data set V corresponds to the set of black pixels containing 2,059 points. The initial condition u_0 is the signed distance function to a circle enclosing V . The Figure 6 (top) shows the resulting solution u at convergence and more specifically the discretization of the zero level set of u obtained using the triangulation T (bottom). The latter contains 8,417 vertices and 16,766 triangles, corresponding to a minimal size $h_{\min} = 2.57 \times 10^{-4}$ for a time step $\Delta t = 10 h_{\min}$. It is interesting to notice that after 120 iterations, most of the shape has already been captured, at the noticeable exception of the *Région Aquitaine* that is fully converged after 850 iterations with an L^2 -error of approximately 3.1×10^{-3} (*cf.* Figure 8, bottom right).

The extraction of a piecewise affine discretization of the zero level set allows to define the boundary of a domain Ω that can be remeshed according to different mesh sizing function more appropriate to numerical applications (like wind modelling and ocean streams analysis in fluid dynamics). Figure 6 (bottom) shows a mesh of Ω (left) and of the complementary part of Ω (right).

6.4.2. Noisy set of points

To show the robustness of our approach, we are considering adding a Gaussian noise to the coordinates of the points of V . To this end, we use the Box-Muller method [8] to generate random numbers, to satisfy a reduced centered normal distribution from numbers verifying uniform law. For a given point P , we consider U_1 and U_2 to independant random

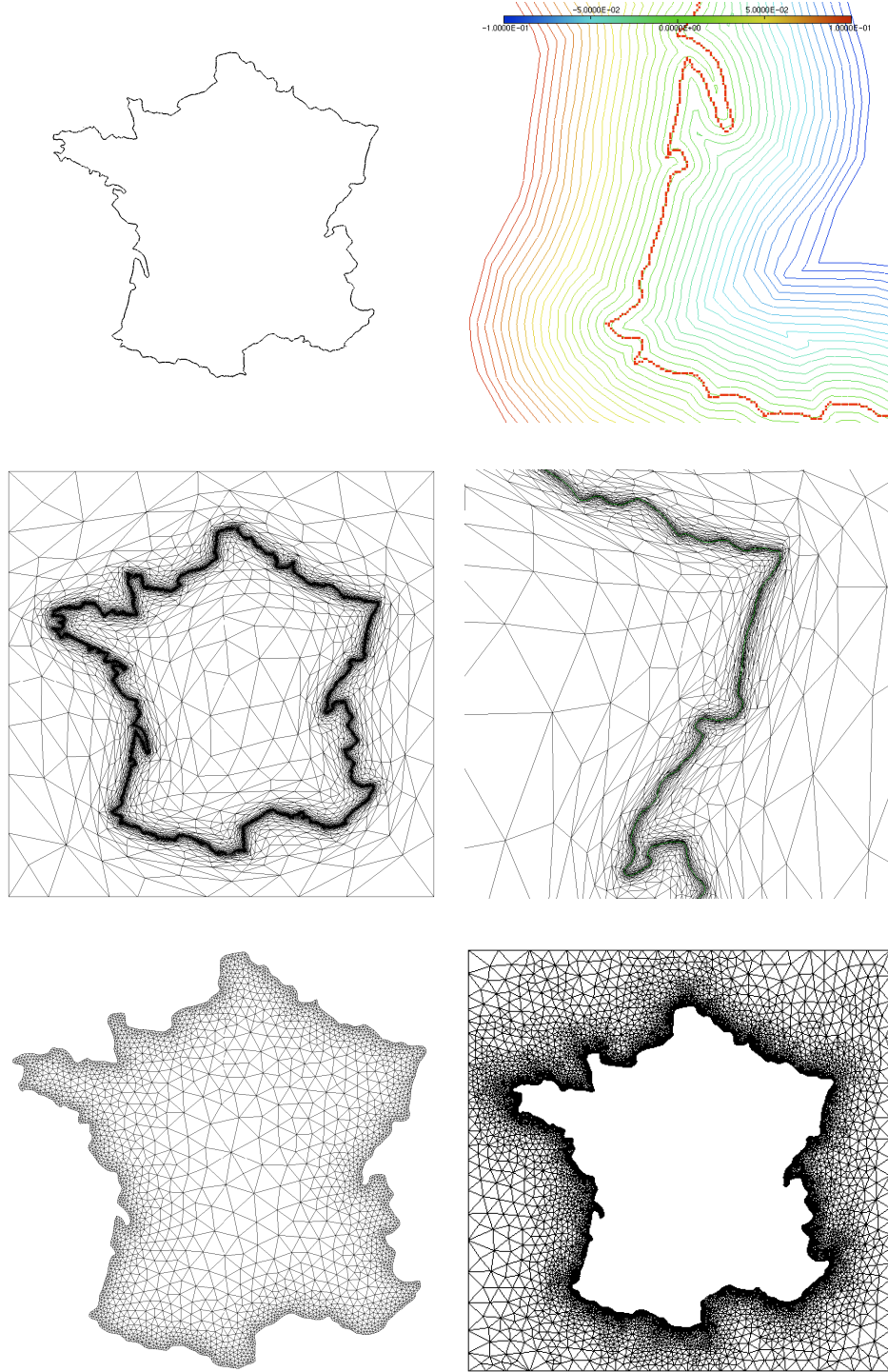


Figure 6: *Application to image processing. Zero level set of u at convergence (top left) and local enlargement in the neighborhood of V (top right). Anisotropic triangulation used for the resolution (middle left) and local enlargement in the vicinity of V (middle right). Examples of computational meshes corresponding to a piecewise affine approximation deduced from the resulting curve (bottom).*

variable uniformly distributed in the range $]0, 1]$. Denoting x, y the coordinates of P , the noisy coordinates (x_n, y_n) are then computed as follows:

$$\begin{cases} x_n = x + \sigma T_1 \\ y_n = y + \sigma T_2, \end{cases}$$

where σ is standard deviation and:

$$\begin{cases} T_1 = \sqrt{-2 \ln U_1} \cos(2\pi U_2) \\ T_2 = \sqrt{-2 \ln U_1} \sin(2\pi U_2), \end{cases}$$

with $\sigma = 2 d_{\min}$, where d_{\min} is the minimal distance between any two points in the initial (non noisy) set V .

Figure (7) shows the corresponding solution at convergence. For the sake of efficiency, we used a different adapted mesh with a larger minimal size of $h_{\min} = 1.2 \times 10^{-3}$, as the need for accuracy is here less important than in the previous test case because of the amplitude of the noise signal. Here, the L^2 - norm of the error at convergence is about 3.9×10^{-2} .

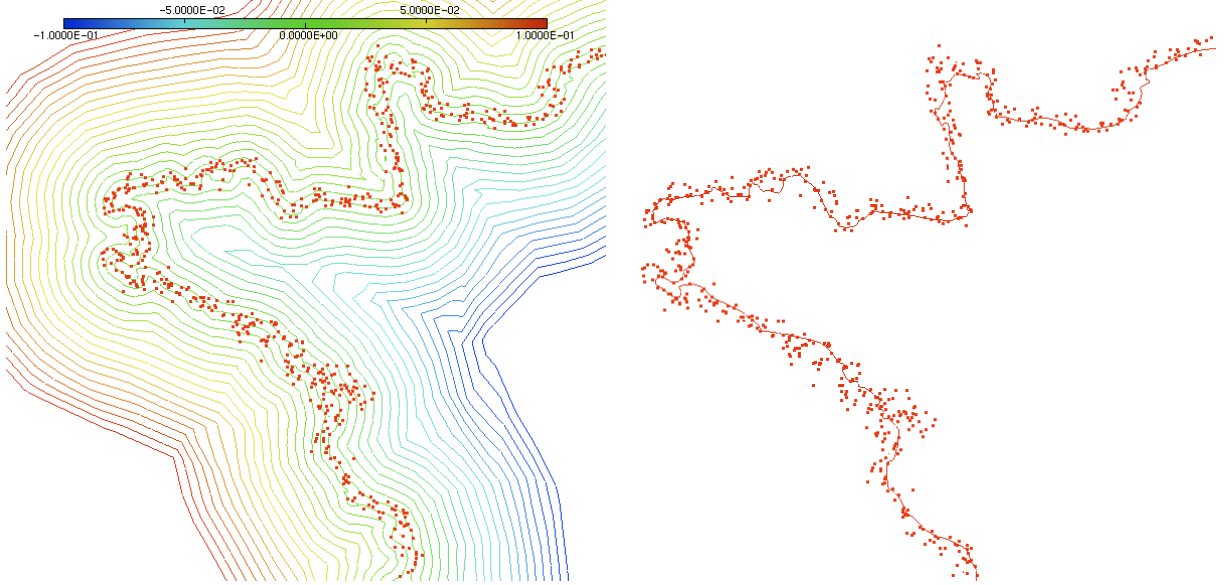


Figure 7: *Regular curve reconstruction on noisy data set. Isovalues of the solution at convergence (left) and zero isovalue of the solution (right).*

6.4.3. Impact of the initial condition

We have pointed out in the introduction that there is no specific constraint on the initial condition. To illustrate this feature, Figure 8 shows the evolutions of the zero level set with respect to two different conditions, corresponding to the situation where the set V is partly or not encompassed by the initial zero level curve. In both cases, the final curve fits at best the data set. 850 iterations have been necessary to obtain the convergence on this dataset (Figure 9, right).

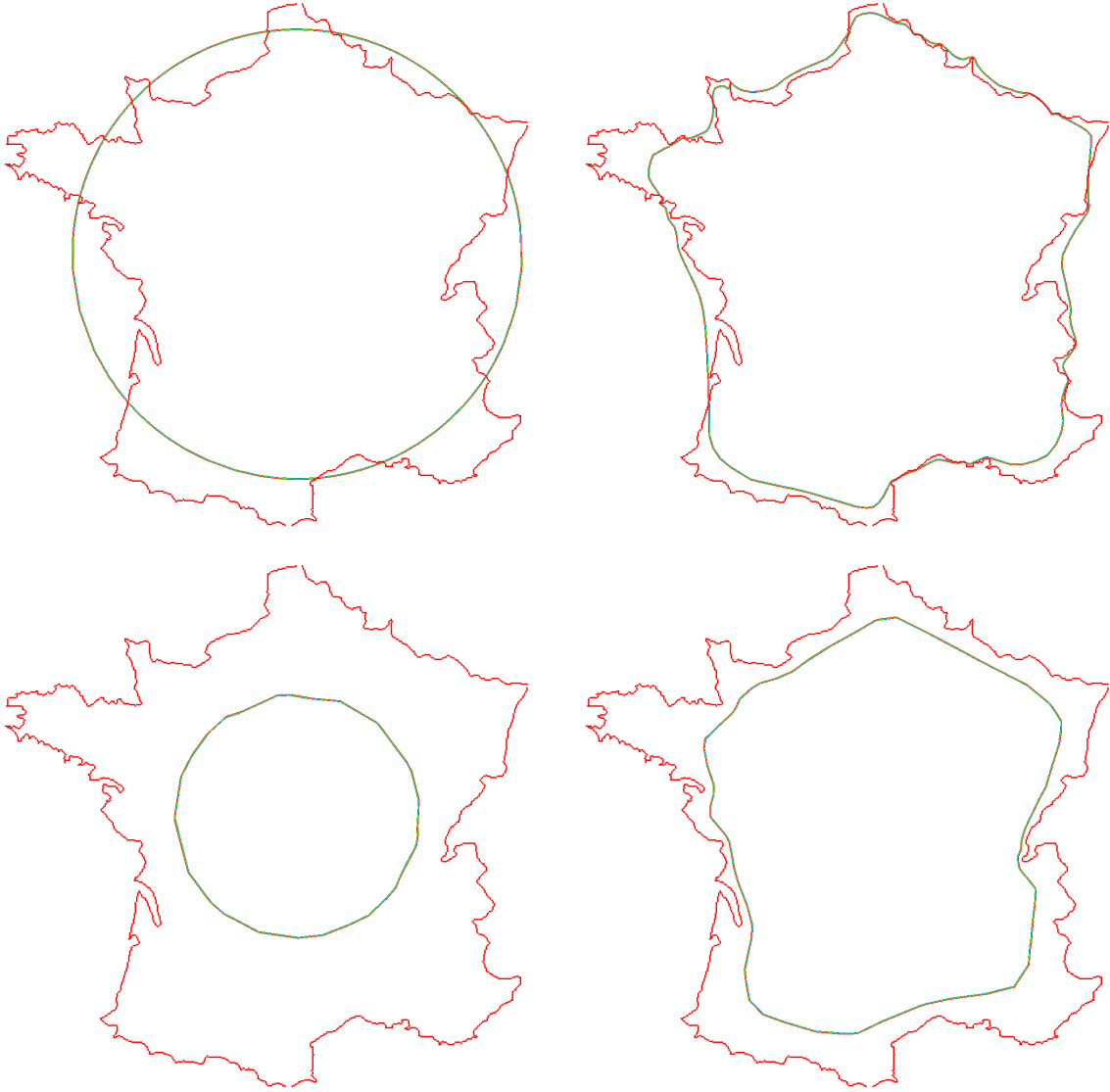


Figure 8: *Impact of the initial conditions (top and bottom left) on the same point set and associated intermediate evolution curves (top and bottom right).*

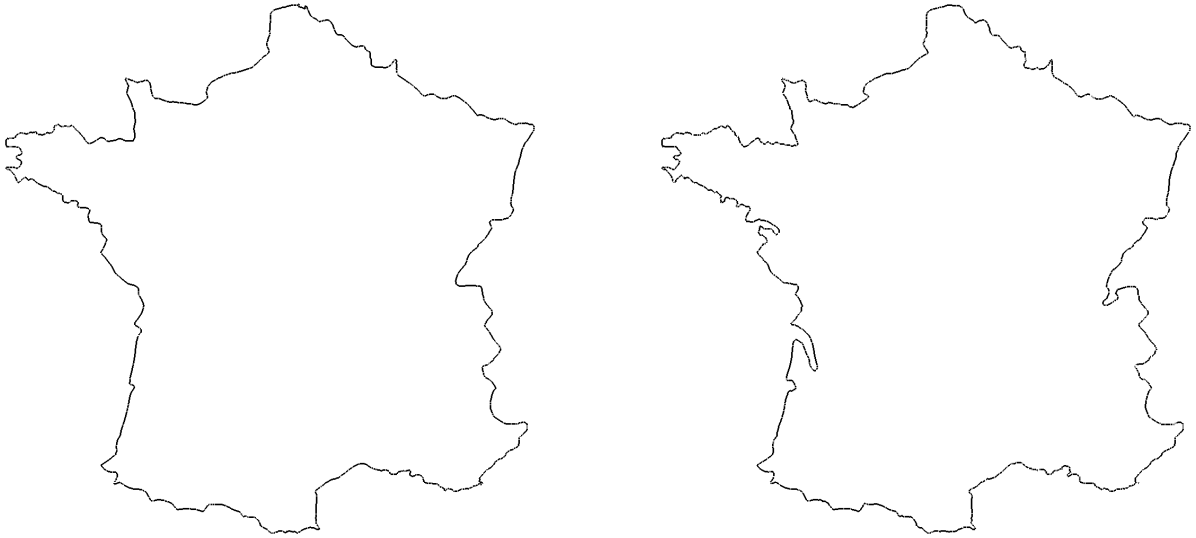


Figure 9: *Curve reconstruction after 120 iterations (left) and at convergence after 850 iterations (right).*

6.4.4. Set of points containing several connected components

To emphasize the efficiency of our method in dealing with several connected components, we consider the data set V containing 6,241 points (Figure 10, top left). We start from an initial signed distance function to a circle enclosing V , *i.e.* defining a single connected component for the zero level set. At completion (Figure 10, left), the several connected components of the domain have all been captured and delimited by a regular curve. The adapted triangulation T_h contains 3,867 vertices and 7,706 triangles corresponding to a minimal size $h_{\min} = 1.2 \times 10^{-3}$ and a time step $\Delta t = 10 h_{\min}$ (*c.f.* Figure 10). The convergence is achieved after 620 iterations with an L^2 -error corresponding to 2.86×10^{-3} .

From these three examples, we observe that the numerical results are in good accordance with the theoretical expectations and we can see that the L^2 -error is of the order of the mesh minimal size h_{\min} .

6.5. Smooth surface reconstruction

Finally, we propose an example of surface reconstruction using our approach. As pointed out, this approach works exactly the same way in three dimensions. The dataset V is represented on Figure 11 (left) with a cut through the anisotropic adapted mesh (right) that contains 293,324 vertices and 1,510,741 tetrahedra. One can easily see the refinement region on the cut plane of the three-dimensional mesh corresponding to the zero level set of the unsigned distance function to the data set. The minimal element size of the triangulation is $h_{\min} = 5 \times 10^{-3}$ and by comparison a uniform mesh of this minimal size would have about 4×10^9 vertices and would eventually not fit into the computer memory.

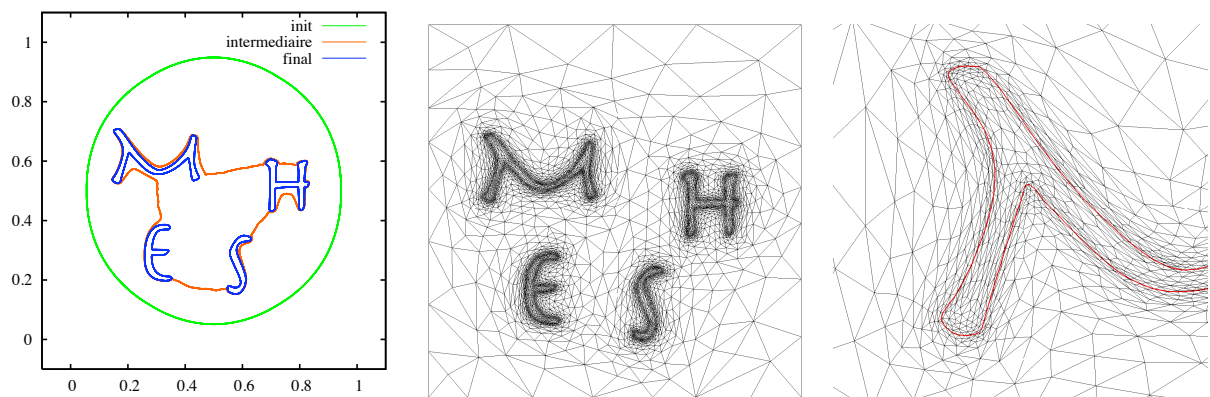


Figure 10: *Curve reconstruction with several connected components.*

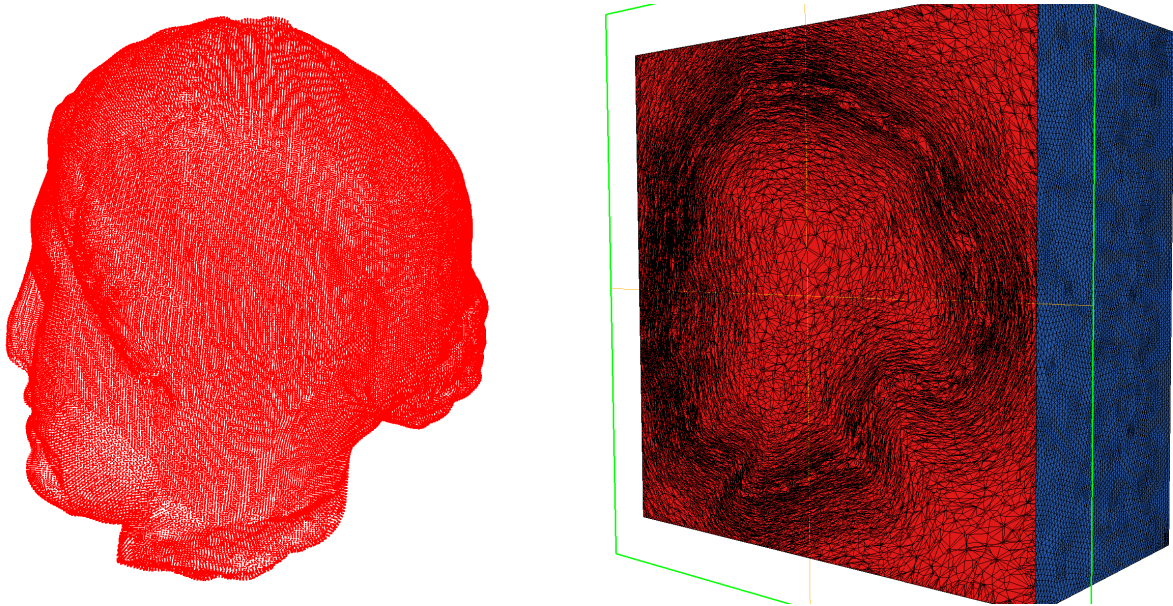


Figure 11: *Data set V for the statue (left) and cut through the adapted anisotropic mesh (right).*

Figure 12 (left) shows a piecewise affine approximation of the zero isosurface obtained by the method on the dataset V containing 134,345 points. On the right part of the figure, several reconstructions are proposed to emphasize the ability of our approach to deal with different sets of sampling points. The purpose of this example is to show that different datasets lead to different surface reconstructions of various resolution. This option is particularly useful in the context of data visualization, where different occurrences of the same surface can be used, the level of accuracy being then related for instance to the distance to the viewer. The hierarchical meshes, on the right of the Figure 12, have respectively 230,000 (bottom), 21,000 (middle) and 12,000 (top) vertices.

7. Conclusion

In this paper, we have proposed a new PDE-based method for reconstructing a smooth and closed manifold surface from a sample of unstructured points. The model has been implemented in two and three dimensions using the level set formalism pioneered by Osher and Sethian for tracking evolutionary surfaces and taking advantage of a semi-implicit scheme we developed and analyzed. Different applications examples have been provided to emphasize the strength of this method. On the theoretical side, the only missing piece is a formal proof of convergence of our scheme for it is lacking monotonicity. Work is still in progress on this topic.

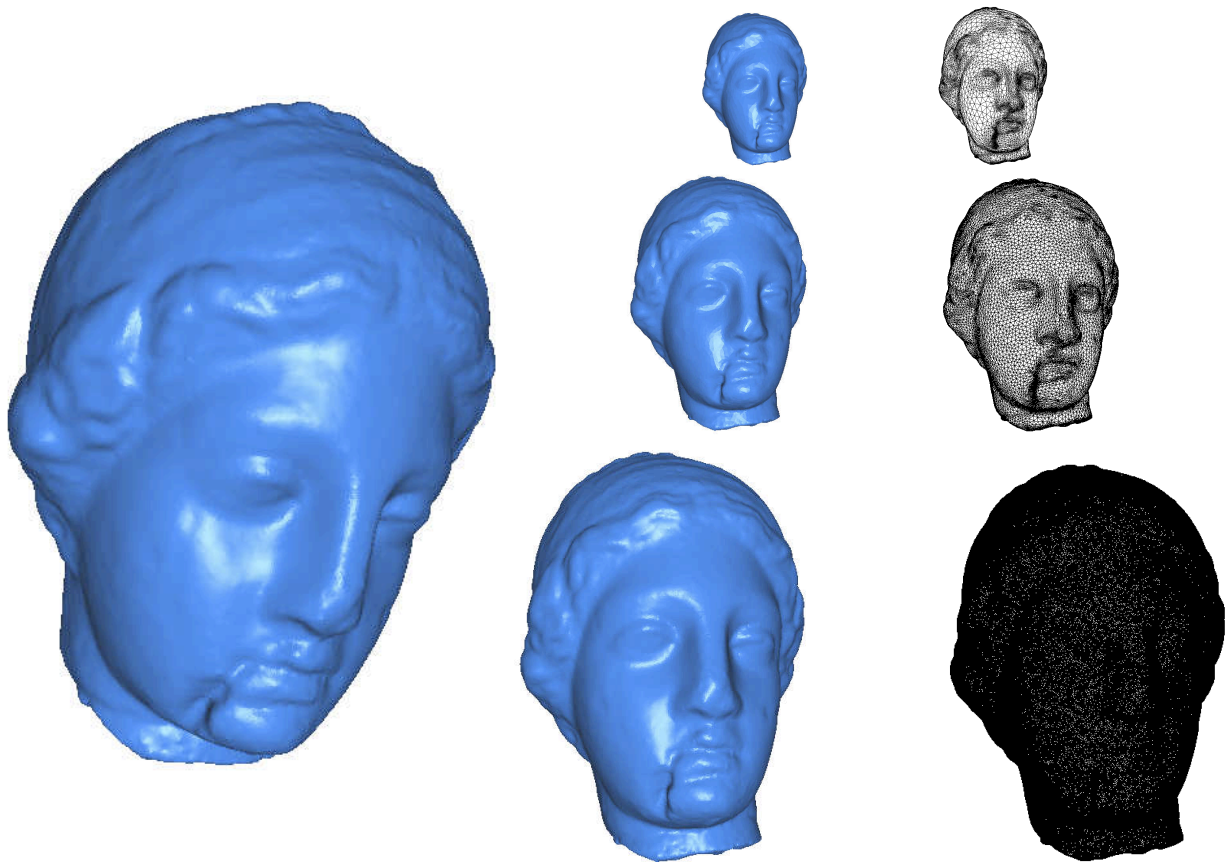


Figure 12: *Example of surface reconstruction. Left: final smooth surface obtained by our approach. Right: a sequence of hierarchical meshes obtained by our approach, corresponding to various data sets.*

Appendix A. Coefficients of the stencil of the approximation of the curvature term

We denote $c_{i,j}$ the coefficient at the point $x_{i,j}$ appearing in front of $u_{i,j}$ in the stencil decomposition of the curvature discrete operator $\nabla \cdot \left(\frac{\nabla}{|\nabla|} \right)_{i,j}$.

$$\begin{aligned}
c_{i,j} &= \frac{4}{a_{i+1,j}} + \frac{4}{a_{i,j+1}} + \frac{4}{a_{i-1,j}} + \frac{4}{a_{i,j-1}} + \frac{2}{a_{i+1,j+1}} + \frac{2}{a_{i-1,j+1}} + \frac{2}{a_{i-1,j-1}} + \frac{2}{a_{i+1,j-1}}, \\
c_{i-1,j-1} &= 0, \quad c_{i,j-1} = \frac{2}{a_{i+1,j-1}} + \frac{2}{a_{i,j-1}} + \frac{2}{a_{i+1,j+1}} + \frac{2}{a_{i,j+1}}, \\
c_{i+1,j-1} &= 0, \quad c_{i+1,j} = \frac{2}{a_{i-1,j}} + \frac{2}{a_{i-1,j-1}} + \frac{2}{a_{i+1,j-1}} + \frac{2}{a_{i+1,j}}, \\
c_{i+1,j+1} &= 0, \quad c_{i,j+1} = \frac{2}{a_{i+1,j}} + \frac{2}{a_{i-1,j}} + \frac{2}{a_{i+1,j+1}} + \frac{2}{a_{i-1,j+1}}, \\
c_{i-1,j+1} &= 0, \quad c_{i-1,j} = \frac{2}{a_{i-1,j+1}} + \frac{2}{a_{i,j+1}} + \frac{2}{a_{i-1,j-1}} + \frac{2}{a_{i,j-1}}, \\
c_{i-2,j-2} &= \frac{-2}{a_{i-1,j-1}}, \quad c_{i-1,j-2} = \frac{-2}{a_{i,j-1}} - \frac{2}{a_{i-1,j-1}}, \\
c_{i,j-2} &= \frac{-4}{a_{i,j-1}}, \quad c_{i+1,j-2} = \frac{-2}{a_{i+1,j-1}} - \frac{2}{a_{i,j-1}}, \\
c_{i+2,j-2} &= \frac{-2}{a_{i+1,j-1}}, \quad c_{i+2,j-1} = \frac{-2}{a_{i+1,j}} - \frac{2}{a_{i+1,j-1}}, \\
c_{i+2,j} &= \frac{-4}{a_{i+1,j}}, \quad c_{i+2,j+1} = \frac{-2}{a_{i+1,j+1}} - \frac{2}{a_{i+1,j}}, \\
c_{i+2,j+2} &= \frac{-2}{a_{i+1,j+1}}, \quad c_{i+1,j+2} = \frac{-2}{a_{i,j+1}} - \frac{2}{a_{i+1,j+1}}, \\
c_{i,j+2} &= \frac{-4}{a_{i,j+1}}, \quad c_{i-1,j+2} = \frac{-2}{a_{i-1,j+1}} - \frac{2}{a_{i,j+1}}, \\
c_{i+2,j} &= \frac{-4}{a_{i+1,j}}, \quad c_{i+2,j+1} = \frac{-2}{a_{i+1,j+1}} - \frac{2}{a_{i+1,j}}, \\
c_{i+2,j+2} &= \frac{-2}{a_{i+1,j+1}}, \quad c_{i+1,j+2} = \frac{-2}{a_{i,j+1}} - \frac{2}{a_{i+1,j+1}}, \\
c_{i,j+2} &= \frac{-4}{a_{i,j+1}}, \quad c_{i-1,j+2} = \frac{-2}{a_{i-1,j+1}} - \frac{2}{a_{i,j+1}}, \\
c_{i-2,j+2} &= \frac{-2}{a_{i-1,j+1}}, \quad c_{i-2,j+1} = \frac{-2}{a_{i-1,j}} - \frac{2}{a_{i-1,j+1}}, \\
c_{i-2,j} &= \frac{-4}{a_{i-1,j}}, \quad c_{i-2,j-1} = \frac{-2}{a_{i-1,j-1}} - \frac{2}{a_{i-1,j}}.
\end{aligned}$$

Appendix B. Proof of the Proposition 5.1

In one dimension of space, we recall that the classical scheme involves only the grid points x_{i-1} , x_i , x_{i+1} and then the curvature formula becomes:

$$\delta_x \left(\frac{\delta_x u}{a} \right)_i = \frac{\left(\frac{\delta_x u}{a} \right)_{i+\frac{1}{2}} - \left(\frac{\delta_x u}{a} \right)_{i-\frac{1}{2}}}{\Delta x} = \frac{1}{h^2} \left(\frac{u_{i+1} - u_i}{a_{i+\frac{1}{2}}} - \frac{u_i - u_{i-1}}{a_{i-\frac{1}{2}}} \right).$$

If the grid points $x_{i+\frac{1}{2}}$ and $x_{i-\frac{1}{2}}$ are not defined, we could use instead an average value,

$$a_{i+\frac{1}{2}} = \frac{1}{2}(a_{i+1} + a_i).$$

With the proposed scheme, the curvature expression reads as follows:

$$\begin{aligned} \delta_x \left(\frac{\delta_x u}{a} \right)_i &= \frac{u_{i+2} \omega'_{s_4, i+2} \omega'_{s_3, i+1}}{a_{i+1}} + \frac{u_i \omega'_{s_3, i} \omega'_{s_3, i+1}}{a_{i+1}} + \frac{u_{i-2} \omega'_{s_1, i-2} \omega'_{s_2, i-1}}{a_{i-1}} + \frac{u_i \omega'_{s_2, i} \omega'_{s_2, i-1}}{a_{i-1}} \\ &= \frac{1}{(2h)^2} \left(\frac{u_{i+2} - u_i}{a_{i+1}} - \frac{u_i - u_{i-2}}{a_{i-1}} \right) \sim \delta_x \left(\frac{\delta_x u}{a} \right), \end{aligned}$$

where $\omega'_{s_p, j}$ denotes the derivative of the barycentric coordinate of the point j in the element s_p . We observe that this approximation is consistent with the classical scheme.

In two dimensions of space, the derivatives are simply written as:

$$\nabla \cdot \left(\frac{\nabla u}{a} \right) = \frac{\partial}{\partial x} \left(\frac{\frac{\partial u}{\partial x}}{a} \right) + \frac{\partial}{\partial y} \left(\frac{\frac{\partial u}{\partial y}}{a} \right),$$

and thus the approximation of the curvature at the grid point (i, j) becomes:

$$\begin{aligned} \nabla \cdot \left(\frac{\nabla u}{a} \right)_{i,j} &= \frac{\left(\frac{\frac{\partial u}{\partial x}}{a} \right)_{i+\frac{1}{2}, j} - \left(\frac{\frac{\partial u}{\partial x}}{a} \right)_{i-\frac{1}{2}, j}}{\Delta x} + \frac{\left(\frac{\frac{\partial u}{\partial y}}{a} \right)_{i, j+\frac{1}{2}} - \left(\frac{\frac{\partial u}{\partial y}}{a} \right)_{i, j-\frac{1}{2}}}{\Delta y} \\ &= \frac{1}{h^2} \left(\frac{u_{i+1, j} - u_{i, j}}{a_{i+\frac{1}{2}, j}} - \frac{u_{i, j} - u_{i-1, j}}{a_{i-\frac{1}{2}, j}} + \frac{u_{i, j+1} - u_{i, j}}{a_{i, j+\frac{1}{2}}} - \frac{u_{i, j} - u_{i, j-1}}{a_{i, j-\frac{1}{2}}} \right). \end{aligned}$$

We split now the analysis of our scheme into several steps, and we consider namely:

- the coefficients of the principal cross in the stencil:

$$\frac{1}{64h^2} \begin{bmatrix} & & 4 & & \\ & & 0 & & \\ 4 & 0 & -16 & 0 & 4 \\ & & 0 & & \\ & & 4 & & \end{bmatrix}$$

that leads to:

$$\begin{aligned}
& \frac{1}{64h^2} \left(\frac{4u_{i,j+2} - 4u_{i,j}}{a_{i,j+1}} - \frac{4u_{i,j} - 4u_{i,j-2}}{a_{i,j-1}} + \frac{4u_{i+2,j} - 4u_{i,j}}{a_{i+1,j}} - \frac{4u_{i,j} - 4u_{i-2,j}}{a_{i-1,j}} \right) \\
&= \frac{4}{64} \times \frac{1}{(2h)^2} \left(\frac{u_{i,j+2} - u_{i,j}}{a_{i,j+1}} - \frac{u_{i,j} - u_{i,j-2}}{a_{i,j-1}} + \frac{u_{i+2,j} - u_{i,j}}{a_{i+1,j}} - \frac{u_{i,j} - u_{i-2,j}}{a_{i-1,j}} \right) \\
&\sim \frac{1}{4} \nabla \cdot \left(\frac{\nabla u}{a} \right),
\end{aligned} \tag{B.1}$$

- the coefficients of the second and fourth columns of the stencil:

$$\frac{1}{64h^2} \begin{bmatrix} 4 \\ 0 \\ -8 \\ 0 \\ 4 \end{bmatrix} \quad \text{and} \quad \frac{1}{64h^2} \begin{bmatrix} 4 \\ 0 \\ -8 \\ 0 \\ 4 \end{bmatrix}$$

leading respectively to:

$$\begin{aligned}
& \frac{1}{64h^2} \left(\frac{2u_{i-1,j+2} - 2u_{i-1,j}}{a_{i,j+1}} - \frac{2u_{i-1,j} - 2u_{i-1,j-2}}{a_{i,j-1}} \right) \\
&+ \frac{1}{64h^2} \left(\frac{2u_{i-1,j+2} - 2u_{i-1,j}}{a_{i-1,j-1}} - \frac{2u_{i-1,j} - 2u_{i-1,j+2}}{a_{i-1,j+1}} \right) \quad \text{and} \\
& \frac{1}{64h^2} \left(\frac{2u_{i+1,j+2} - 2u_{i+1,j}}{a_{i,j+1}} - \frac{2u_{i+1,j} - 2u_{i+1,j-2}}{a_{i,j-1}} \right) \\
&+ \frac{1}{64h^2} \left(\frac{2u_{i+1,j+2} - 2u_{i+1,j}}{a_{i+1,j+1}} - \frac{2u_{i+1,j} - 2u_{i+1,j-2}}{a_{i+1,j-1}} \right),
\end{aligned} \tag{B.2}$$

- the coefficients of the second and fourth rows:

$$\frac{1}{64h^2} \begin{bmatrix} 4 & 0 & -8 & 0 & 4 \end{bmatrix} \quad \text{and} \quad \frac{1}{64h^2} \begin{bmatrix} 4 & 0 & -8 & 0 & 4 \end{bmatrix}$$

leading respectively to:

$$\begin{aligned}
& \frac{1}{64h^2} \left(\frac{2u_{i+2,j+1} - 2u_{i,j+1}}{a_{i+1,j}} - \frac{2u_{i,j+1} - 2u_{i-2,j+1}}{a_{i-1,j}} \right) \\
&+ \frac{1}{64h^2} \left(\frac{2u_{i+2,j+1} - 2u_{i,j+1}}{a_{i+1,j+1}} - \frac{2u_{i,j+1} - 2u_{i-2,j+1}}{a_{i-1,j+1}} \right) \quad \text{and} \\
& \frac{1}{64h^2} \left(\frac{2u_{i+2,j-1} - 2u_{i,j-1}}{a_{i+1,j}} - \frac{2u_{i,j-1} - 2u_{i-2,j-1}}{a_{i-1,j}} \right) \\
&+ \frac{1}{64h^2} \left(\frac{2u_{i+2,j-1} - 2u_{i,j-1}}{a_{i+1,j-1}} - \frac{2u_{i,j-1} - 2u_{i-2,j-1}}{a_{i-1,j-1}} \right).
\end{aligned} \tag{B.3}$$

One can easily observe that: (B.2) + (B.3) $\sim \frac{1}{2} \nabla \cdot \left(\frac{\nabla u}{a} \right)$.

- the remaining coefficients:

$$\frac{1}{64h^2} \begin{bmatrix} 2 & & & 2 \\ & 0 & & 0 \\ & & -8 & \\ & 0 & & 0 \\ 2 & & & 2 \end{bmatrix}$$

corresponding to:

$$\begin{aligned} \frac{1}{64h^2} \left(\frac{2u_{i+2,j-2} - 2u_{i,j}}{a_{i+1,j-1}} - \frac{2u_{i,j} - 2u_{i+2,j+2}}{a_{i+1,j+1}} + \frac{2u_{i-2,j+2} - 2u_{i,j}}{a_{i-1,j+1}} - \frac{2u_{i,j} - 2u_{i-2,j-2}}{a_{i-1,j-1}} \right) \\ \sim \frac{1}{4} \nabla \cdot \left(\frac{\nabla u}{a} \right). \end{aligned} \quad (\text{B.4})$$

Then, by summing (B.1), (B.2), (B.3) and (B.4), we deduce that our scheme is consistent with the operator $\nabla \cdot \left(\frac{\nabla}{a} \right)$.

Appendix C. Proof of the Proposition 5.2

In this appendix, we show that the matrix $(I - \Delta t D M)$ is invertible. To this end, we want to show that $\text{Ker}(I - \Delta t D M) = \{0\}$. We suppose there exists $u \neq 0$, such that $(I - \Delta t D M)u = 0$. We introduce the diagonal matrix Λ such that $(\Lambda)_{i,i} = |\mathcal{B}_i|$. We have then $(I - \Delta t D \Lambda^{-1} \Lambda M)u = 0$, and $(\Lambda D^{-1} - \Delta t \Lambda M)u = 0$. Hence, we obtain $u^T (\Lambda D^{-1} - \Delta t \Lambda M)u = 0$ and

$$u^T \Lambda D^{-1} u = \Delta t u^T \Lambda M u. \quad (\text{C.1})$$

As $u^T \Lambda D^{-1} u > 0$ if $u \neq 0$, to assure the invertibility of $(I - \Delta t D M)$, let us prove that ΛM is a symmetric negative semi-definite matrix in order to have $\text{Ker}(I - \Delta t D M) = \{0\}$.

First of all, ΛM is symmetric, *i.e.* $\langle \Lambda M u, v \rangle = \langle u, \Lambda M v \rangle$.

Indeed, using Equation (14) we recall that:

for $\alpha_{i_2} = |\nabla u|_{i_2}$ then,

$$\left(\nabla \cdot \left(\frac{\nabla u}{|\nabla u|} \right) \right)_i = \frac{\sum_{K \in \mathcal{B}_i} |K| \sum_{x_{i_2} \in K} \left\langle \frac{(\nabla u)_{i_2}}{\alpha_{i_2}}, \nabla \omega_{K,i_2} \right\rangle}{|\mathcal{B}_i|}.$$

We introduce the diagonal matrix Λ such that $(\Lambda)_{i,i} = |\mathcal{B}_i|$. Thus, we write:

$$\begin{aligned}\langle \Lambda M u, v \rangle &= \sum_{x_i} (\Lambda M u)_i v_i = \sum_{x_i} v_i \sum_{K \in \mathcal{B}_i} |K| \left(\nabla \cdot \left(\frac{\nabla u}{|\nabla u|} \right) \right)_i \\ &= \sum_{x_i} v_i \sum_{K \in \mathcal{B}_i} |K| \sum_{x_{i_2} \in K} \left\langle \frac{(\nabla u)_{i_2}}{\alpha_{i_2}}, \nabla \omega_{K,i_2} \right\rangle \\ &= \sum_{x_i} \sum_{K \in \mathcal{B}_i} \sum_{x_{i_2} \in K} v_i |K| \left\langle \frac{(\nabla u)_{i_2}}{\alpha_{i_2}}, \nabla \omega_{K,i_2} \right\rangle.\end{aligned}$$

Using a renumbering procedure, $(K \in \mathcal{B}_i \Rightarrow x_i \in K)$ and $(x_{i_2} \in K \Rightarrow K \in \mathcal{B}_{i_2})$ we obtain :

$$\langle \Lambda M u, v \rangle = \sum_{x_{i_2}} \left\langle \frac{(\nabla u)_{i_2}}{\alpha_{i_2}}, \sum_{K \in \mathcal{B}_{i_2}} |K| \sum_{x_i \in K} v_i \nabla \omega_{K,i_2} \right\rangle.$$

The next step is to prove that, for all x_{i_2} :

$$\sum_{K \in \mathcal{B}_{i_2}} |K| \sum_{x_i \in K} v_i \nabla \omega_{K,i_2} = - \sum_{K \in \mathcal{B}_{i_2}} |K| \sum_{x_i \in K} v_i \nabla \omega_{K,i},$$

i.e., for all x_{i_2} :

$$\sum_{K \in \mathcal{B}_{i_2}} |K| \sum_{x_i \in K} v_i (\nabla \omega_{K,i_2} + \nabla \omega_{K,i}) = 0 \quad \Leftrightarrow \quad \sum_{x_i \in \mathcal{B}_{i_2}} \sum_{K \in \mathcal{B}_{i_2} \cap \mathcal{B}_i} v_i |K| (\nabla \omega_{K,i_2} + \nabla \omega_{K,i}) = 0. \quad (\text{C.2})$$

It is sufficient to show that for all x_{i_2} , $\sum_{K \in \mathcal{B}_{i_2} \cap \mathcal{B}_i} |K| (\nabla \omega_{K,i_2} + \nabla \omega_{K,i}) = 0$, $\forall x_i \in \mathcal{B}_{i_2}$. This

condition is also a necessary condition, for the equality (C.2), to be satisfied for all v_i . We have to consider two cases.

- If $x_i = x_{i_2}$, the same term $\nabla \omega_{K,i_2}$ appears twice in the previous sum.

We define $\omega = \sum_{K \in \mathcal{B}_{i_2}} \omega_{K,i_2} \mathbb{I}_K$, and thus we have:

$$\sum_{K \in \mathcal{B}_{i_2}} |K| \nabla \omega_{K,i_2} = \int_{\bigcup_{K \in \mathcal{B}_{i_2}} K} \nabla \omega = \int_{\partial(\cup K)} \langle \omega, \vec{n} \rangle = 0.$$

This term is vanishing since $\omega = 0$ on $\partial(\cup K)$. And hence the result follows:

$$\sum_{K \in \mathcal{B}_{i_2}} |K| \nabla \omega_{K,i_2} = 0.$$

- If $x_i \neq x_{i_2}$, the sum becomes:

$$\sum_{K \in \mathcal{B}_i \cap \mathcal{B}_{i_2}} |K| (\nabla \omega_{K,i_2} + \nabla \omega_{K,i}) = |K| (\nabla \omega_{K,i_2} + \nabla \omega_{K,i}) + |L| (\nabla \omega_{L,i_2} + \nabla \omega_{L,i}),$$

where K and L denote the two triangles sharing the edge (x_i, x_{i_2}) , and (k, l) the two vertices opposed to the edge (x_i, x_{i_2}) in K, L respectively.

Since $\sum_{x_{i_2} \in K} \nabla \omega_{K i_2} = 0$ for all x_{i_2} , we have $\nabla \omega_{K i_2} + \nabla \omega_{K i} = -\nabla \omega_{K k}$ and $\nabla \omega_{L i_2} + \nabla \omega_{L i} = -\nabla \omega_{L l}$.

Considering the two points opposite to the edge (x_i, x_{i_2}) in K and L , we like to show that:

$$|K| \nabla \omega_{K k} = -|L| \nabla \omega_{L l}.$$

We notice that $\nabla \omega_{K k}$ and $\nabla \omega_{L l}$ are orthogonal to the edge (x_i, x_{i_2}) . Moreover, $|\nabla \omega_{K k}| = \frac{1}{h_k}$, where h_k is the height of K emanating from k and $|K| = \frac{h_k}{2} \times |(x_i, x_{i_2})|$, where $|(x_i, x_{i_2})|$ denotes the length of the edge (x_i, x_{i_2}) , we conclude that $|K| |\nabla \omega_{K k}| = \frac{|(x_i, x_{i_2})|}{2}$. Likewise, on the triangle L , we obtain a similar relation: $|L| |\nabla \omega_{L l}| = \frac{|(x_i, x_{i_2})|}{2}$. Note that the vectors $\nabla \omega_{K k}$ and $\nabla \omega_{L l}$ have the same directions but opposite signs. Hence, $|K| \nabla \omega_{K k} = -|L| \nabla \omega_{L l}$ and the results (C.2) follows.

We have just shown that:

$$\sum_{K \in \mathcal{B}_{i_2}} |K| \sum_{x_i \in K} v_i \nabla \omega_{K i_2} = - \sum_{K \in \mathcal{B}_{i_2}} |K| \sum_{x_i \in K} v_i \nabla \omega_{K i}. \quad (\text{C.3})$$

On the other hand, we have the following equality:

$$\sum_{K \in \mathcal{B}_{i_2}} |K| \sum_{x_i \in K} v_i \nabla \omega_{K, i_2} = - \sum_{K \in \mathcal{B}_{i_2}} |K| (\nabla v)_{i_2},$$

and then, we obtain the identity:

$$\begin{aligned} \langle \Lambda M u, v \rangle &= \sum_{x_{i_2}} \left\langle \frac{(\nabla u)_{i_2}^2}{\alpha_{i_2}}, - \sum_{K \in \mathcal{B}_{i_2}} |K| (\nabla v)_{i_2} \right\rangle \\ &= - \sum_{x_{i_2}} \left\langle \frac{(\nabla u)_{i_2}}{\alpha_{i_2}}, (\nabla v)_{i_2} \right\rangle \sum_{K \in \mathcal{B}_{i_2}} |K| \\ &= - \sum_{x_{i_2}} \left\langle \frac{(\nabla v)_{i_2}}{\alpha_{i_2}}, (\nabla u)_{i_2} \right\rangle \sum_{K \in \mathcal{B}_{i_2}} |K| \\ &= \langle u, \Lambda M v \rangle. \end{aligned} \quad (\text{C.4})$$

The matrix ΛM is then symmetric. Moreover, by taking $u = v$ in (C.4), we have

$$\langle \Lambda M u, u \rangle = - \sum_{x_{i_2}} \frac{(\nabla u)_{i_2}^2}{\alpha_{i_2}} \sum_{K \in \mathcal{B}_{i_2}} |K| \leq 0.$$

Thus ΛM is a symmetric negative semi-definite matrix.

By considering again the relation (C.1), we recall that $u^T D \Lambda^{-1} u > 0$ if $u \neq 0$ and we have just shown that ΛM is a symmetric negative semi-definite matrix, then the relation (C.1) is impossible. This leads to write $\text{Ker}(I - \Delta t DM) = 0$ and $(I - \Delta t DM)$ is invertible.

References

- [1] R. Abgrall, On essentially non-oscillatory schemes on unstructured meshes: analysis and implementation, *J. Comp. Phys.*, **114**, (1994), 45-58.
- [2] G. Allaire, Conception optimale de structures, coll. Mathématiques et Applications, vol. 58, *Springer*, (2005).
- [3] N. Amenla, M. Bern and M. Kamvysselis, A new Voronoï based surface reconstruction algorithm, *ACM SIGGRAPH Proc.*, (1998).
- [4] D. Attali, R-regular shape reconstruction from unorganized points, *ACM Symp. Computational Geometry*, (1997), 248-253.
- [5] G. Barles, P.E. Souganidis, Convergence of approximation schemes for fully nonlinear second order equations, *Asymptotic analysis*, **4** (1991), 271-283.
- [6] F. Bernardini and C. Bajaj, Sampling and reconstructing manifolds using alpha-shapes, *Proc. 9th Canadian Conf. Computational Geometry*, (1997), 248-253.
- [7] J.D. Boissonnat and F. Cazals, Smooth shape reconstruction via natural neighbor interpolation of distance functions, *ACM Symposium on Computation Geometry*, (2000), 223-232.
- [8] G.E.P. Box and M.E. Muller, A note on the generation of random normal deviates, *The annals of mathematical statistics*, **29**, (1958), 610611.
- [9] R. Bridson, J. Teran, N. Molino and R. Fedkiw, Adaptive physics based tetrahedral mesh generation using level sets, *Engineering with computers*, **21**, (2005), 2-18.
- [10] J.C. Campbell, J.M. Hyman, M.J. Shashkov, Mimetic finite difference operators for second order tensors on unstructured grids, *Computers and Mathematics with Applications*, **44**, (2002), 157-173.
- [11] C. Caselles, F. Catte, T. Coll and F. Dibos, A geometric model for active contours in image processing, *Numer. Math.*, **66**, (1993), 1-31.
- [12] Y.G. Chen, Y. Giga and S. Goto, Uniqueness and existence of viscosity solutions of generalized mean curvature flow equations, *Proc. Japan Acad. Ser. A*, **65**, 1989.
- [13] A. Claisse, Modèle de reconstruction d'une surface échantillonnée par une méthode de ligne de niveau, et applications, Thèse de doctorat, UPMC, (2009).

- [14] A. Claisse and P. Frey, Construction d'une courbe régulière d'approximation d'un ensemble de points, *C. R. Acad. Sci., Ser. I*, **346**, (2008), 1017-1022.
- [15] A. Claisse and P. Frey, Level set driven smooth curve approximation from unorganized or noisy point sets, *ESAIM: Proceedings*, **27**, (May 2009), 254-271.
- [16] D. Cohen-Steiner and F. Da, A greedy Delaunay-based surface reconstruction algorithm, *The visual computer*, **20**, (2004), 4-16.
- [17] M.G. Crandall and P-L. Lions, Viscosity solutions of Hamilton-Jacobi equations, *Trans. Amer Math Soc*, **277**, (1983), 1-42.
- [18] M.G. Crandall, H. Ishii and P.L. Lions, User's guide to viscosity solutions of second order partial differential equations, *Bull. AMS*, **27/1**, (1992), 1-67.
- [19] T. Dey and J. Sun, Normal and feature approximations from noisy point clouds, *Proc. FST & TCS, Lecture Notes in Computer Science, Springer*, **4337**, (2006), 21-32.
- [20] Y. Duan, L. Yang, H. Qin and D. Samaras, Shape reconstruction from 3D and 2D data using PDE-based deformable surfaces, *Proc. European Conference on Computer Vision*, **3**, (2004), 238-251.
- [21] V. Ducrot and P. Frey, Contrôle de l'approximation géométrique d'une interface par une métrique anisotrope, *C. R. Acad. Sci. Paris, Ser. I*, **345**, 537-542.
- [22] R. Dyer, Surface reconstruction, *APMA 990, Final project*, (2003).
- [23] L.C. Evans, Partial differential equations, AMS (1997).
- [24] L. C. Evans and J. Spruck, Motion of level sets by mean curvature, I., *J. Differential geometry*, **33**, (1991), 635-681.
- [25] P. Frey and P.-L. George, Mesh generation: application to finite elements, *Wiley-ISTE*, (2008).
- [26] P.L. George and H. Bourouchaki, Triangulation de Delaunay et maillages, Hermès (1997).
- [27] H. Henrot and M. Pierre, Variation et optimisation de formes. Une analyse géométrique, coll. Mathématiques et Applications, vol. 48, *Springer*, (2005).
- [28] H. Hoppe, T. de Rose, T. Duchamp, J. Mc Donald and W. Stuetzle, Surface reconstruction from unorganized points, *Proc. Siggraph*, (1992), 71-78.
- [29] H. Ishii and M.-H. Sato, Nonlinear oblique derivative problems for singular degenerate parabolic equations on a general domain, *Nonlinear Analysis*, **57**, (2004), 1077-1098.

- [30] P.D. Lax, R.D. Richtmyer, Survey of the stability of linear finite difference equations, *Comm. Pur. Appl. Math.*, **9**, (1956), 267-293.
- [31] J. Liang, T. Mc Inerney and D. Terzopoulos, United Snakes, *Medical image analysis*, **10**, (2006), 215-233.
- [32] W.E. Lorensen and H.E. Cline, Marching cubes : a high resolution 3D surface construction algorithm, *Computer Graphics*, **21**, (1987), 163-169.
- [33] D. Lu, H. Zhao, M. Ziang, S. Zhou and T. Zhou, A surface reconstruction method for highly noisy point clouds, *Proc. Variational, geometric, and level set methods in computer vision*, **3752**, (2005), 283-294.
- [34] P. Macklin and J. Lowengrub, An improved geometry-aware curvature discretization for levelset methods : Application to tumor growth, *J. Comp. Phys.*, **215**, (2006), 392-401.
- [35] R. Mencl and H. Müller, Interpolation and approximation of surfaces from three-dimensional scattered data points, *Scientific Visualisation*, (1997), 223-232.
- [36] E.E. Okon, Finite difference approximations for the three-dimensional Laplacian in irregular grids, *J. App. Math. and Physics.*, **33**, (1982), 266-281.
- [37] S. Osher and J. Sethian, Front propagating with curvature dependent speed : algorithms based on Hamilton-Jacobi formulations, *J. Comp. Physics*, **79**, (1988), 12-49.
- [38] S. Osher and N. Paragios, Geometric Level Set Methods in Imaging, Vision and Graphics, *Springer*, (2003).
- [39] A. Sard, The measure of the critical values of differentiable maps, *Bull. Am. Math.*, **48**, (1943), 883-890.
- [40] I. Semenova, V. Savchenko and I. Hagiwara, Reconstruction of shapes based on normals analysis, *Proc. GraphiCon*, (2005), 1-7.
- [41] Y. Shi and W.C. Karl, Shape reconstruction from unorganized points with a data-driven level set method, *Proc. IEEE Int. Conf. on Acoustics, Speech and signal processing*, (2004), 13-16.
- [42] M. Sussman, P. Smereka and S. Osher, A level set method for computing solutions to incompressible two-phase flows, *J. Comp. Phys.*, **114**, (1994), 146-159.
- [43] M. Sussman and E. Fatemi, An efficient interface preserving level set redistancing algorithm and its application to interfacial incompressible fluid flow, *SIAM J. Sci. Comp.*, **20**(4), (1999), 1165-1191.
- [44] H.K. Zhao, S. Osher, B. Merriman and M. Kang, Implicit and non-parametric shape reconstruction from unorganized data using a variational level set method, *Comput. Vision and Image Understanding*, **80**, (2000), 295-314.

- [45] H. Zhao and S. Osher, Visualization, analysis and shape reconstruction of unorganised data sets, in geometric level set methods in imaging, *Vision and graphics*, S. Osher and N. Paragios eds., Springer, (2002).